COMPUTER SCIENCE

For Class

9th - 10th

Part-1

Punjab Textbook Board, Lahore.

Approved by: Federal Ministry of Education, Curriculum Wing Islamabad
vide its letter no. F-1/2005-Maths(Comp.s)dated 16-02-2005

Contents

Authors

☆ **Mirza Mubbasher Baig**
Research Associate (Computer Science)
Lahore University of Management Sciences (LUMS). Lahore.

☆ **Rana Azeem Muhammad Khan**
Lecturer (Computer Science)
Govt. College University, Lahore.

Editors

☆ **Syed Zulqarnain Jaffery**
Assistant Professor
COMSATS Institute of Information Technology, Sector H-8 Islamabad.

☆ **Asif Ali**
Lecturer, Govt. Model College for Boys, G-3, Islamabad.

☆ **Dr. Muhammad Naeem Gul**
Associate Professor, University of Engineering & Technology, Lahore.

Supervision :

Mr. Mazhar Hayat
Subject Specialist, Punjab Text Book Board, Lahore.

Publisher: Punjab Textbook Board, Lahore.
Printer: Majid Book Depot, Lahore.

Date of Printing	Copies
Jan 2013	505000

Chapter 1

Introduction to Computers

Today, you can find **computers** almost everywhere. Microwave ovens, automobiles, thermostats and even wristwatches contain computer chips. In fact, computers are so common in modern society that everyone is affected by computer technology to some extent.

A computer is an electronic device that processes **data** and converts it into information. Computers run **programs**, which process data and perform tasks based on the instructions contained in the program. Because a computer program can evaluate data and then based on those evaluations, results are generated which can be used for large number of applications. Computers can process data with little assistance. Some examples for computer application are:

- Controlling space flight
- Landing airplanes
- Tracking inventory
- Printing books
- Turning on lights at a specified time
- Checking out groceries at the checkout counter

In this chapter, we will explore the history of computers and define the different types of computers available today. Also, the impact of computers on society will be discussed alongwith the introduction to programming languages.

1.1 History of Computers

The history of computers starts out about 5000 years ago, with the birth of the **abacus.** It is a wooden rack holding horizontal wires with beads strung on them. When these beads are moved around, according to programming rules memorized by the user, all regular arithmetic problems can be done.

Figure 1.1: A simple ABACUS

1.1.1 Napier's Bones

John Napier, a Scottish mathematician, created logarithm tables to facilitate calculations. He also created a device using rods, also called Napier's Bones to perform arithmetic calculations. These rods were widely used by accountants and bookkeepers. Several people used the concept of logarithms to develop the slide rule. With a modern slide

rule you could not only perform the arithmetic operations, but also calculate squares, square roots, logs, sine, cosine, and tangent. The slide rule was used till the middle 70's.

1.1.2 Pascal's Pascaline Calculator

Pascal invented a machine that had a system of gears. A one-tooth gear engages its single tooth with a ten-tooth gear once every time it revolves. It must make ten revolutions to rotate the ten-tooth gear once. Numbers could be entered and cumulative sums obtained by cranking a handle. Pascal's calculator was not a commercial success because these devices could not be built with sufficient precision for practical use.

The German mathematician, **von Leibniz**, produced a machine that was similar to Pascal's but more reliable and accurate. Other mechanical calculators followed that were refinements on the designs of Pascal and Leibniz.

1.1.3 Charles Babbage

While Thomas of Colmar was developing the calculator, a series of very interesting development in computers was started in Cambridge, England by Charles Babbage, a mathematics professor. He began to design an automatic mechanical calculating machine, which he called a **difference engine**. By 1822, he had a working model to demonstrate. It was intended to be steam powered and fully automatic, including the printing of the results. Babbage continued to work on it for the next 10 years, but in 1833 he lost interest because he thought he had a better idea i.e. the construction of what would now be called a general purpose, fully program-controlled, automatic mechanical digital computer. Babbage called this idea an **Analytical Engine**. The idea of this design showed a lot of foresight, although this couldn't be appreciated until a full century later. The machine was supposed to operate automatically, by steam power, and require only one person.

1.1.4 Use of Punched Cards by Hollerith

In 1890, Herman Hollerith developed the first electro-mechanical punched card tabulator. The tabulator could read information that had been punched into cards. These cards were maintained in stack form. Solutions to different problems could be stored on different stacks of cards and accessed when needed.

Invention of punched cards opened a gate to modern data processing. IBM and other computer manufacturers came forward and started production of punched-card using computers. These computers could add, multiply and sort numbers. Data were fed and results were produced on punched cards.

As compared to today's machines, these computers were slow, usually processing 50 - 220 cards per minute, each card holding about 80 decimal numbers (characters). At that time, however, punched cards were a huge step forward. They provided a means for Input/Output (I/O), and memory storage on a huge scale.

1.1.5 Electronic Digital Computers

The start of World War II produced a large need for computer capacity, especially for the military. New weapons were made for which large number of calculations was needed. In 1942, John P. Eckert, John W. Machly, and their associates at the Moore school of Electrical

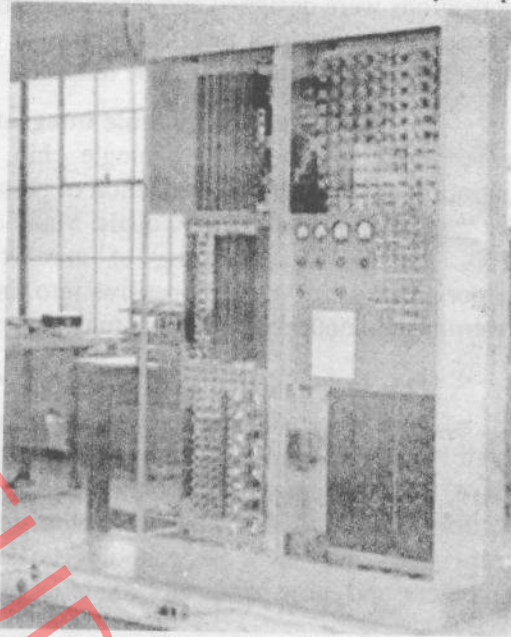
Engineering, University of Pennsylvania decided to build a high speed electronic computer to do the job. This machine was known as ENIAC (Electrical Numerical Integrator And Calculator).

1.1.6 The Modern Stored Program EDC

Fascinated by the success of ENIAC, the mathematician John Von Neumann in 1945 undertook a study of computation that showed that a computer should have a very simple, fixed physical structure, and yet be able to perform any kind of computation without the need for any physical change in the unit.

Figure 1.2: EDVAC

Von Neumann contributed a new awareness of how practical and fast computers should be built. These ideas, usually referred to as the **stored - program** technique, became essential for future generations of high speed digital computers and were universally adopted. According to Von Neumann theory *"Data and program can be stored in the same memory. Thus the machine can itself alter either its program or internal data"*.

As a result of these ideas, computing and programming became much faster, more flexible, and efficient.

This group of computers included EDVAC (Figure 1.2) and UNIVAC, which were the first commercially available computers.

1.1.7 Advancement in 1950's – 1960's

In early 1950's, two important engineering inventions changed the image of the computer field. These discoveries were the magnetic core memories and the Transistor Circuit Elements. This quickly found its way into new models of digital computers.

These machines were very expensive and were also complicated to operate. Such computers were mostly found in large computer centers, government, and research and development laboratories. Those computers mostly worked on a single problem at a time. During this period, the major computer manufacturers began to offer a range of computer equipment with different prices, as well as accessories such as:

- Card Readers
- Printers
- Cathode-Ray-Tube

These were widely used in businesses for such things as:

- Accounting
- Payroll
- Inventory control
- Ordering Supplies
- Billing

Central Processing Units (CPUs) for these usages did not have to be very fast and were usually used to access large amount of records on a computer file. The computer systems were sold for applications, such as hospitals, banks, defense etc.

1.1.8 More Recent Advancements

The trend during the 1970's was moving away from very powerful single purpose computers toward a larger range of applications for cheaper computer systems. New applications were designed and made available for controlling a great range of manufacturing processes. Moreover, a new revolution in computer hardware was under the way, which allowed the size of computer to be reduced.

In 1980's, **very large scale integration** (VLSI), in which hundreds of thousands of transistors were placed on a single chip, became more and more common. The trend continued and led to the introduction of personal computers (PCs), which are smaller in size, inexpensive and used by individuals. Many companies introduced very successful PCs in the 1970s. In the manufacturing of computer processor chips, the Intel and Motorola Corporations were very competitive into the 1980s. In early 1980s, however, the Japanese government announced a big plan to design and build a new generation of computers.

This new generation, the so-called "fifth" generation, is using new technologies and will be capable of amazing features such as artificial intelligence. The cost of computers is rapidly lessening, and their convenience and efficiency are expected to increase in the early future. The computer field continues to experience huge growth. Computer networking, electronic mail, and electronic publishing are just a few of the applications that have grown in recent years. Advances in technologies continue to produce cheaper and more powerful computers and now computers are present in most of the homes, offices, and schools.

1.2 Computer Generations

1.2.1 First Generation –Vacuum Tubes

Computers of this generation used vacuum tubes (Figure 1.3) to perform calculations. **Vacuum tubes** were expensive because of the amount of material and skill needed to manufacture them. Vacuum tubes get hot and burn out. Computers of this generation were very large machines. Special rooms with air conditioning were needed to house them because of the heat generated by the vacuum tubes. The most important computers were ENIAC and UNIVAC – I.

Figure 1.3: Vacuum Tube

ENIAC (Electronic Numerical Integrator and Calculator)

It was the first general-purpose electronic digital computer designed by John William Machly and John Eckert in 1942. The ENIAC was very heavy and large in size. It consumed 140 kilowatts of power and was capable of doing 5000 additions

per second. ENIAC was a decimal rather than a binary machine. That is numbers were represented in decimal form and arithmetic was performed in the decimal system. The major drawback of ENIAC was that it had to be programmed manually by setting switches and plugging and unplugging cables.

UNIVAC (Universal Automatic Computer)

In 1947, Eckert and Machly formed Eckert-Machly Computer Corporation to manufacture computers commercially. Their first successful machine was UNIVAC, which was delivered to US bureau of census in 1951. It was actually the first computer developed for commercial use. It was intended for both scientific and commercial applications.

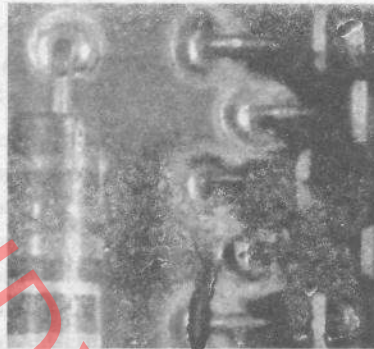
1.2.2 Second Generation – Transistors

TRANSISTORS

Transistor was invented in 1947 by William Shockley, John Bardeen, and William Brattain.

Figure 1.4 Transisters

Advantages

- 200 transistors are about the same size as one vacuum tube in a computer.
- Much less expensive than a vacuum tube.
- A transistor can work 40 times faster than a vacuum tube.
- Do not get hot and burn out like a vacuum tube.

The major change in the electronic computer was the replacement of vacuum tubes with transistors. The transistor was invented at Bell Labs in 1947. The transistor is smaller, cheaper, and dissipates less heat than a vacuum tube but can be used in the same way as a vacuum tube to construct computers.

As compared to first generation computers, second generation computers were smaller and had high processing speed. Most of these computers used magnetic core memory as internal storage. The second-generation computers enjoyed the use of more complex arithmetic and logic unit, use of low level and high level programming languages such as COBOL, BASIC, Pascal and Assembly etc. and provision of system software with the computers. Examples of second-generation computers are IBM 7094 series, IBM 1400 series, CDC 164 etc.

1.2.3 Third Generation - Integrated Circuits

IC (Integrated Circuit)

1. The concept of the IC was developed by Jack St. Clair Kilby in 1958.
2. First IC was invented and used in 1961.
3. An IC is about 1/4 square inch and can contain thousands of transistors

The major invention of third generation of computers was the development of IC (integrated circuit). A single IC chip contains thousands of transistors. The computer became smaller in size, faster, more reliable, and lower in price. And also became very common in medium to large-scale business. These computers used magnetic core memory as internal storage. The most successful computers of this generation were IBM system/360 and DEC PDP-8, the others were UNIVAC 1108, UNIVAC 9000 and IBM 370 etc.

1.2.4 Fourth Generation - Microprocessors

MICROPROCESSOR

1. The microprocessor is a complete processing circuitry on a chip. Ted Hoff produced the first microprocessor in 1971 for Intel, which was named as "Intel 4004 ".
2. Modern microprocessors are usually less than one square inch and can contain million of electronic circuits.
3. Used in many electronic devices today such as wristwatches, microwave ovens and cars.

Fourth generation of computers started with the invention of microprocessors. It revolutionized the computer world. Advancements were made in the integrated circuit technology. LSI (Large Scale Integrated Circuits) and VLSI (Very Large Scale Integrated Circuits) were designed which contributed to the invention of microprocessor. Computers of this generation used semiconductor memory, which increased the internal storage of computers. These computers have high processing speed, more internal storage and are smaller in size. Examples of fourth generation computers are apple Macintosh and IBM PC etc.

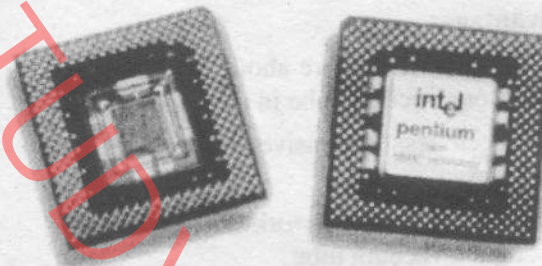



Figure 1.5: Microprocessors

1.2.5 Fifth Generation – Present and Beyond: Artificial Intelligence

Fifth generation computing devices, based on artificial intelligence, are still in development stage.Though there are some applications,such as voice recognition,that are being used today. The use of parallel processing and superconductors is helping to make artificial intelligence a reality. Quantum computation and molecular and nano-technology will radically change the face of computers in years to come. The goal of fifth-generation computing is to develop devices that respond to natural language input and are capable of learning and self-organization.

1.3 Types of Computers

There are three types of computer:

- Analog Computers
- Digital Computers
- Hybrid Computers

1.3.1 Analog Computers

Analog computers use electronic or mechanical phenomena to model the problem being solved by using one kind of physical quantity to represent another.

The analog computer excels in solving dynamic problems and simulating complex physical systems. It has no way for man-machine interaction, recording and graphic display. High-speed computing elements, used to simulate mathematical functions, physical systems, and dynamic processes, are arranged in as much the same pattern as the physical system they represent.

Early special-purpose analog computers were the slide rule, the curvimeter and plainmeter, and the harmonic analyzer. In World War II, analog computing mechanisms were of great importance for gunfire control on warships. General-purpose analog computers were first built in 1930s.

1.3.2 Digital Computers

Digital computers process data in numerical form using digital circuits. The digital computers perform arithmetic and logic operations with discrete values. Digital computers are good at solving algebraic equations and even better at manipulating numbers. It is unbeatable for high-speed precision in arithmetic operations, data storage, and documentation. It can perform only one calculation at a time.

The results can be obtained in a variety of forms, such as printed tables, magnetic tape, and the familiar punched cards. While properly used in high-accuracy, high-volume numerical calculations, the digital computers are extremely good.

In early 1940s, Aiken built the first general-purpose digital computer called **Mark-1**. With the invention of digital computers, a new age of computing started. Today digital computers are widely being used in business, educational institutes, hospitals etc. for various purposes. Examples of digital computers are IBM PC, Apple's Macintosh computers etc.

1.3.3 Hybrid Computers

Hybrid Computers are the combination of analog and digital computers. Hybrid computers use analog-to-digital conversion and digital-to-analog conversion, and may input or output either analog or digital data. Dynamic problems that once took too long or were too difficult to handle can be solved in a reasonable period of time. These computers can produce highly accurate and precise results. These types of computers are used in robotics, medical labs etc.

1.4 Classification of Computer

Computers come in many different sizes and ranges of power, and different types of computer systems have varying capabilities. Today's computer systems are classified into following categories:

- Super Computer
- Mainframe Computer
- Mini Computers
- Micro computer

1.4.1 Supercomputers

Supercomputers are the most powerful computers made, and physically they are some of the largest. These systems are built to process huge amount of data, and the fastest supercomputers can perform more than 1 trillion calculations per second. Some supercomputers – such as Cray T90 system – can house thousands of processors. This speed and power make supercomputers ideal for handling large and highly complex problems that require extreme calculating power e.g. these computers help analyze and forecast global weather patterns, nuclear scientists use these computers to perform complex calculations etc.

Supercomputers can cost tens of millions of dollars and consume enough electricity to power dozens of homes. Because of their size and cost, supercomputers are relatively rare, used only by large corporations, universities, and government agencies that can afford them.

1.4.2 Mainframe Computers

The largest type of computer in common use is the mainframe. **Mainframe** computers are used in large organizations like insurance companies and banks where many people need frequent access to the same data, which is usually organized into one or more huge database. Airlines use large mainframe systems for flight scheduling, reservations, ticketing, and meeting a range of customer service needs.

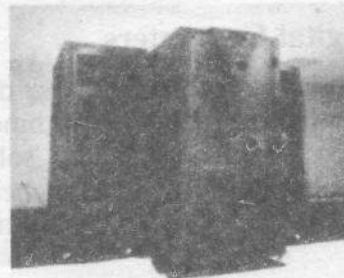



Figure. 1.6 Mainframe Computer

In traditional mainframe environment, each user works at a computer terminal. A **terminal** is a monitor and a keyboard connected to a mainframe. These computers are larger in size, expensive, can store huge amount of data (billions of records) and can support thousands of terminals. These computers are basically used in networked environment, as a single user can not utilize their full processing power. IBM S/390 is an example of mainframe computers.

1.4.3 Minicomputer

Minicomputers got their name because of their small size. These computers have less processing power than mainframe computers but have high processing power than microcomputers. Like mainframe, minicomputers can support number of user's input and output requirements. Normally minicomputers are used in networked environment as server machines. These computers are less expensive than mainframe computes and are ideal for organizations that could not afford mainframe or do not need the processing power of mainframe computers. The HP 3000 is an example of minicomputer.

1.4.4 Microcomputer

Microcomputers (also referred to as personal computers) are typically developed for individual users. These are less powerful machines as compared to minicomputers. In 1981, IBM called its first microcomputer the IBM-PC. Within a few years many other manufacturers copied the architecture and IBM compatible computers emerged in market.

One big factor of the popularity of microcomputers is the low price. PCs are getting powerful day by day because of the improvement in technology. That's why the difference between more powerful microcomputers and less powerful minicomputers is vanishing. The most powerful PCs are about as much powerful as a less powerful minicomputer can be. Individuals are using microcomputers for performing certain tasks. These are also used in business, education, and almost every field of life.

Microcomputers are available in different models i.e. desktop models, laptop computers, and pocket computers etc.

POCKET (PALMTOP) COMPUTER

Pocket computers have been designed to allow people to keep lots of information close to hand wherever they happen to be. A pocket computer has to have small, light batteries that last a long. These computers have special operating systems suited to pocket computers. One problem with small computers is that they don't have full-sized keyboards attached. These computers use special pens and touch-sensitive screens to enter data as well as a number of small buttons or keys.

Figure 1.7: Palmtop Computers

LAPTOP COMPUTER

The main aim of a laptop is that the persons using it can have all programs and data from their desktop computer on a portable computer.

The person using a laptop should be able to run all the same software on the laptop as runs on larger, desktop computers as laptop computers have the same types of operating system as desktop ones. Modern laptops can have floppy drives, CD-ROM drives and CD re-writers, and even DVD drives. They often have full-sized keyboards and a mouse or a touch-sensitive mouse pad. The screen is usually a large Liquid Crystal Display (LCD).

Figure 1.8: Laptop Computer

Laptops are usually much more expensive than desktop computers. They have expensive battery packs that have to power the hard disk, CD drives and LCD screen.

The batteries generally don't last as long as those in a pocket computer and may need recharging more than once a day, depending on the use.

DESKTOP COMPUTER

At the moment there are two main types of desktop computer available.

1. The Macintosh (made by Apple Computers)
2. Personal Computers (PC,s)

Macintosh is usually distinguished by its stylish look and bright colors (Figure 1.9).

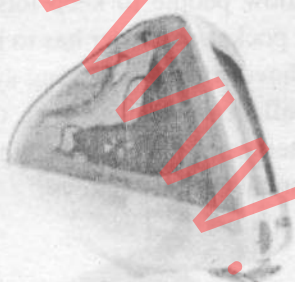



Figure 1.9: Macintosh Computer

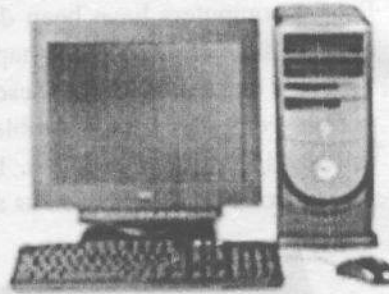



Figure 1.10 : IBM Compatible PC

When people talk about PCs (Figure 1.10) they usually mean an IBM-compatible computer based on an Intel (or similar) microprocessor. The most common operating system for the PC is Microsoft Windows (latest version Windows XP) although other operating systems are available (e.g. Linux).

1.5 Impact of Computers and Internet on Society

There are numerous areas in which the increased ease and efficiency of computers have put a large impact. Probably the most important on our society is on information sharing.

EDUCATION

Educational institutes, from primary to university level, are using computers for various learning activities. A large number of learning programs (tutorials) are available on almost every subject. These provide a one-to-one interaction with the student. The trend of conducting online examinations is getting popular e.g. GRE, GMAT, SAT etc. are conducted online through out the world. Questions are marked by the computer, which minimize the chance of committing mistakes. It also makes it possible to announce results in time.

Distance learning is a new learning methodology. Computer plays the key role in this type of learning. Hundreds of institutes are offering distance-learning programs. The students are not required to come to the institute instead they are provided reading material and attend classes via **virtual classroom.** In a virtual classroom, the teacher deliver lecture at its own workplace while the students, connected to a network, may listen to him at their homes. They may put questions and answers are sent to them via email.

BUSINESS

Computer is now being widely used in business and industry. Computer information systems are used to keep track of huge transactions. They also allow transactions to be made from anywhere in the world. These are also used to control machines which manufacture products, keep track of customer's bills, analyze sales of various products in different localities on monthly and yearly basis; calculating and recording employ pays and performs various other tasks. These are also widely used in business community to reduce the administrative paperwork and cost.

ONLINE BANKING

The advent of the Internet and the popularity of personal computers presented an opportunity for the banking industry. For years, banking institutions have used powerful computers to perform millions of transactions. Nowadays, ATMs are installed everywhere; these are all computerized and connected together. These can be used to draw money from any branch of that bank at any time of the day. Customers are now also connected to the bank via personal computers, which allows them to see their bank account status at home.

Banks view computerized banking as a powerful tool to attract and retain new customers while eliminating costly paper handling and increasing competitive banking environment. Following are some of the advantages of online banking

- **Convenience:** Computerized online banking sites never close; they're available 24 hours a day, seven days a week, and they can be accessed from a computer.
- **Ubiquity:** If you're out of the country and a money problem arises, you can log on instantly to your online bank and make appropriate transactions.
- **Transaction speed:** Online bank sites generally execute and confirm transactions at quicker processing speed.
- **Efficiency:** You can access and manage all of your bank accounts, from one site.

APPLICATION IN RETAILING APPLICATIONS

Modern stores are quickly incorporating computer system for a number of reasons. Firstly these systems allow the billing of items to be done at great speed. They accept credit cards, allowing customer to purchase goods without cash.

The items at store are marked with "Bar Code". This is called as "Universal Production Code". This is a sequence of lines, which is read by a "Bar Code Reader". The price of the item is stored in this code and these are automatically added to the bill. The computer generates the receipt and the customer pays the bills.

Figure 1.11: Universal Production Code

The computer along with generating the bills also updates the inventory list. This allows the store manager to see that which items are short and in more demand. Marketing experts can also use this information.

COMPUTER SIMULATIONS

Computer Simulation is the use of computer to represent the dynamic responses of one system by the change in behaviors of another system modeled after it. Computer simulations are widely used in educational institutes to make clear the understanding of the working of various systems e.g. simulation of aeroplane is a part of training of the pilot. It makes him /her clear the working of various parts of aeroplane. Simulation of river systems can be manipulated to determine the potential effects of dams, and irrigation network before any actual construction has taken place.

In educational institutes, simulations also have advantages over hands-on laboratory work such as allowing students to do more complicated and hazardous experiments, obtain results more quickly, and get a deeper understanding of the experiments. Simulations can include simple graphical or numerical representations of how chemical or physical experiments are carried out.

APPLICATION IN ENTERTAINMENT

The advancement of computer science has also helped entertainment industry. Nowadays computer can be used to watch television shows being broadcasted on Internet, watch movies, listen to music and play games.

Computers are also used to create animations and special effects for television shows, commercials, movies and cartoons. These allow them to add graphics and colorful displays in their shows.

Computer games are also becoming a big attraction for children and kids. The computers capability to portray graphics is constantly improving. Thus computer games are getting better and better each day. Computer games can take us into the excitement of competition by presenting 3D color images interactively. They can display colors and life like characters, animations, sounds and even videos.

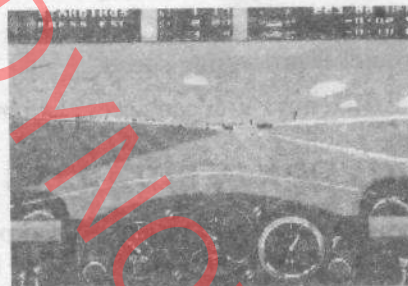



Figure1.12: A car race computer game

Computer is also being used by music industry to create high quality music and sounds in less amount of time. Computerized electronic synthesizers are being used to store, modify and access a wide variety of sounds. New software are allowing musicians to create composition with less effort.

OTHER AREAS

Today computers are being used in many other areas to save time and cost. These include publishing where documents can be written and saved on a computer. This is accomplished by a word-processing application such as Microsoft Word. These applications allow the writers to correct and print the document in very short time. These documents can also be sent from one place to anther via Internet.

Computers are also being used in libraries to maintain and track record of books and library members. Any information regarding the book's author or date of issue can be searched from the computer in seconds. They can also be issued warning when the books issued to them are over-due and must be returned immediately.

Thus computer has allowed people to work fast, saving time and money. Also it has helped in spreading information and knowledge via Internet. The future will allow many additional opportunities for the applications of automation to many areas of life.

1.6 Introduction to Programming Languages

Computer can perform various tasks depending upon the requirements of the user. In order to perform these tasks, computers need instructions which tell them how to execute the required task. *The set of instructions (executed by the CPU) to solve particular problem is called computer program (or simply program).*

A programming language defines a format for writing instructions in a specific order, which are to be executed by a computer. Programming languages also provide a framework for expressing algorithms. Programming languages are a way of communicating with the computer. The actual details of how programming concepts work are not so simple however, we will discuss them briefly.

1.6.1 Types of Computer Languages

There are many different computer languages available for writing programs. Each has its strengths and weaknesses and must be assessed based upon need. A language that is particularly well suited for one application may not work for another.

There are two types of computer languages:

- Low Level Languages
- High Level Languages

LOW LEVEL LANGUAGES

Low level languages provide the programmer with a high degree of control, but they require a detailed knowledge of the hardware to be used. They are really only required for advanced programming needs. There are two main types of low level languages.

- Machine language
- Assembly language

MACHINE LANGUAGE

The processor within a computer can perform various operations, each of which is identified by an operation code (or opcode). It is possible to write a program directly in machine code by using the correct opcodes in the correct sequence into memory, alongwith the required data values and parameter values. The program could be depicted as a series of binary numbers. This however is not a very practical way to write a program. Apart from being complex and time-consuming, programs

written in this way would tend to be error prone and would be very difficult to debug. For this reason, programs are generally written in a language, which is easier for humans to understand and can also be translated into machine code for the processor to understand.

ASSEMBLY LANGUAGE

Assembly language is very close to machine language. The commands are represented in Assembly Language by short names called mnemonics (pronounced as Ne-Monics). For example ld means Load Accumulator with a particular data value. Because each type of processor has a different set of operations, different processors use different Assembly languages. Assembly language programming is complex but it provides a much higher degree of control than high level languages. Programs written in Assembly Language code are translated into machine code by an assembler. Machine code can also be converted back into assembly code using a deassembler.

HIGH LEVEL LANGUAGES

High level languages are close to human languages and far from the machine language. These are machine independent languages which are also known as "third generation" languages. These languages consist of English words, basic mathematical symbols and a few punctuation characters. These languages allow simple statements to be expressed concisely. Each high level language has its own language translater The history of some of the major programming languages is given below.

- **FORTRAN (FORmula TRANslation)**
  In 1957, FORTRAN appeared as the first major high level languages appeared in the form of FORTRAN. FORTRAN stands for Formula Translation. The language was designed at IBM for scientific computing. It was mainly used for scientific purposes.

- **BASIC (Beginners All-purpose Symbolic Instruction Code)**
  BASIC was designed to allow students to write programs using time-sharing computer terminals. BASIC was intended for easy learning programming concepts. The design principles of BASIC were:
  - Be easy for beginners to use.
  - Be a general-purpose language.
  - Allow advanced features to be added for experts (while keeping the language simple for beginners).
  - Be interactive.
  - Provide clear and friendly error messages.
  - Respond fast for small programs.
  - Not require an understanding of computer hardware.
  - Shield the user from the operating system.

- **COBOL (Common Business Oriented Language)**

Though FORTRAN was good at handling numbers, it was not so good at handling input and output, which mattered most to business computing. COBOL was designed as the language for businessmen. Its only data types were numbers and strings of text. A COBOL program consists of four or five major sections. COBOL statements also have a very English-like grammar, making it quite easy to learn. Because of its simplicity it had been very popular among business community.

- **LISP**

LISP stands for List Processing language. It was designed for Artificial Intelligence (AI) research. Because it was designed for such a highly specialized field, its syntax (programming rules) was very different from ordinary languages.
It has the unique ability to modify itself, and hence grow on its own.
It is being used even today because it is highly specialized.

- **PASCAL (PASCAL is a language named after a Scientist Pascal)**

Pascal was designed in a very orderly approach; it combined many of the best features of the languages in use at the time, COBOL, FORTRAN, and ALGOL. While doing so, many of the irregularities of these languages were cleaned up, which helped it to gain popularity. The combination of features, input/output and solid mathematical features, made it a highly successful language.

- **C and C++**

C was developed in 1972 by Dennis Ritchie while working at Bell Labs. C is very commonly used to program operating systems such as UNIX, Windows, and Macintosh OS etc. It is also very useful for compiler writing.

C++ was an extension to C using OOP (Object Oriented Programming) concept, while maintaining the speed of C. C++ is most often used in simulations, such as games. It is the language of choice in today's Computer Science courses.

- **Visual Basic**

It was the first visual development tool from Microsoft, and it was designed to compete with C, C++, Pascal and any other well known programming languages. When it came out, Visual Basic wasn't very successful. It wasn't until Microsoft released VB 2.0 in 1993 that people really started to discover the power of the language, and when Microsoft released VB 3.0 it had become

the fastest growing programming language in the market. By now, Visual Basic has earned itself the status of a professional programming language.

VB is most often used today to create quick and simple interfaces to other Microsoft products such as Excel and Access without needing a lot of code, though it is possible to create full applications with it.

- **JAVA**
  Sun Microsystems began developing a language with the primarily purpose to control microprocessors used in consumer items such as cable receivers, VCR's, toasters, and also for personal data assistants (PDA). Java offers powerful capabilities of network programming, Internet applications and GUI (Graphical User Interface).

1.7 Introduction to Language Translators

Language translators are the programs that translate a high or level language program into machine code.

Program written in any language is first analyzed by a special piece of software. This piece of software is designed by the language designer. The software checks the program for errors, optimizes the code and generates machine language for that program. All the software available for this purpose can be categorized into three main categories.

- Assembler
- Compiler
- Interpreter

1.7.1 Assembler

An assembler is a program that translates an assembly language program into machine code.

1.7.2 Compiler

A compiler is a program that translates a source program (written in some high-level programming language) into machine language (or machine code). A compiler first reads the whole program before executing it.

1.7.3 Interpreter

An interpreter on the other hand, looks at each line of the program, decides what that line means, checks it for possible errors and then executes that line. If one of the lines is executed repeatedly, it must be scanned and analyzed each time, greatly slowing down the solution of the problem at hand.

Exercise

1. Describe Charles Babbage work in the history of computers?
2. Describe the advancements in the computers during the 1950's and 1960's.
3. Write a note on different computer generations, briefly explaining their features.
4. What is the difference between a digital and an analog computer?
5. Write short notes on the following:
   a. Pocket Computers
   b. Laptop Computers
   c. Micro Computers
6. Describe the Impact of computers and internet on society.
7. Define Computer. Briefly describe classification of computers.
8. Modern computers are based on stored program concept. Who introduces the concept? Discuss his/her contribution in the history of computers.
9. Briefly name and describe some of the applications of computers.
10. What is the difference between low level and high level languages?
11. Discuss some negative aspects of the use of Internet in our society.
12. What is a compiler and an interpreter?
13. Write a note on the following:
    a. Visual Basic
    b. LISP
    c. C/C++
14. Fill in the blanks.
    (i) A computer is an electronic device that processes ____________ and converts it into information that people can use.
    (ii) Blaise Pascal is usually credited for building the first ________ computer in 1642
    (iii) John von Neumann developed the idea of the ______________
    (iv) The DOS operating system was developed by __________
    (v) ________________ computers are a combination of analog and digital
    (vi) When people talk about _____________ they usually mean an 'IBM-compatible' computer based on an Intel (or similar) microprocessor.
    (vii) __________ was the major invention of third generation of computers.
    (viii) ENIAC stands for __________
    (ix) Cray T90 is an example of ____________
    (x) Java is a(n) ______________ language
15. Mark as True or False
    (i) The history of computers starts out about 2000 years ago, with the birth of the abacus.
    (ii) Charles Babbage is usually credited for building the **first digital computer** in 1642.

(iii) Charles Babage began to design an automatic mechanical calculating machine, which he called a difference engine.

(iv) The discoveries that changed the image of computer field were the magnetic core memory and the Transistor Circuit Element.

(v) First generation computers used transistors instead of vacuum tubes for performing calculations.

(vi) FORTRAN was the most popular language of first generation of computers.

(vii) LISP is used in Artificial Intelligence.

(viii) **Slide-rule** is an example of analog computers

(ix) Assembler is a program that assembles windows commands

(x) First of all Apple's Macintosh computer introduces GUI

(xi) Supercomputer does not need any instruction to perform a task

16. Choose the correct option:

(i) Which of the following is not a high level language?

a. Fortran **b.** Basic **c.** C and C++ **d.** Assembly Language
e. Visual Basic

(ii) Which of the following is not true about Personal Computers?

a. The PC was introduced by IBM in 1981.
b. Uses the DOS or Windows operating system developed by Microsoft Corporation.
c. They are analogue machines. **d.** Computers are easy to use.
e. People could do work at home that would be transferable to the company's computer.

(iii) The third generation computers used

a. Vacuum Tubes **b.** Integrated circuits **c.** Transistors **d.** Microprocessors

(iv) A terminal consists of

a. a keyboard, mouse and printer **b.** keyboard and monitor
c. mouse and monitor **d.** system unit and I/O devices

(v) Minicomputer is

e. faster than microcomputer **f.** expensive than micro computer
g. smaller in size than microcomputer **h.** a and b **i.** a and c

Answers

Q.14
(i) data (ii) Digital (iii) Stored-program (iv) IBM (v) Hybrid (vi) Personal Computer
(vii) Integrated Circuit (viii) Electronic Numerical Integrator and Calculator (ix) Supercomputer
(x) High level programming language

Q.15
(i) T (ii) F (iii) T (iv) T (v) F
(vi) F (vii) T (viii) T (ix) F (x) T

Q.16
(i) d (ii) c (iii) B (iv) b (v) d

Chapter
2

Computer Components

A computer is a device that accepts data and processes it for some result based on a sequence of instructions. The sequence of instructions given to computer for manipulating data is known as a ***program***. Computers also include the means for storing data and the program on internal memory. The process of performing arithmetic and logic operations with the help of computer is known as ***electronic data processing (EDP)***.

Figure 2.1: A Computer System (PC)

2.1 Components of Computer System

There are two basic components of a computer system.

- Computer Hardware.
- Computer Software.

2.1.1 Computer Hardware

The physical parts of the computer system that you can touch and feel are known as ***computer hardware***. In a broader sense a computer can be divided into following hardware units:

- Input Unit
- Output Unit
- System Unit

Chapter
2

Computer Components

A computer is a device that accepts data and processes it for some result based on a sequence of instructions. The sequence of instructions given to computer for manipulating data is known as a ***program***. Computers also include the means for storing data and the program on internal memory. The process of performing arithmetic and logic operations with the help of computer is known as ***electronic data processing (EDP).***

Figure 2.1: A Computer System (PC)

2.1 Components of Computer System

There are two basic components of a computer system.

- Computer Hardware.
- Computer Software.

2.1.1 Computer Hardware

The physical parts of the computer system that you can touch and feel are known as ***computer hardware***. In a broader sense a computer can be divided into following hardware units:

- Input Unit
- Output Unit
- System Unit

INPUT UNIT

Input unit of a computer system consists of input devices. Due to the diverse nature of the data, variety of input devices exist to input data in different forms e.g. keyboard is used to enter textual data, mouse is used as a pointing device and to trigger different commands in different applications, microphone is used to enter voice data, and scanner is used to enter image data etc. Input unit of a computer may contain all or some of the above described devices, and even can have many other.

OUTPUT UNIT

Output unit of a computer system consists of output devices. As data presented to the user can take variety of forms, therefore different output devices are needed e.g. monitor is used to display text and images on a screen, printer is used to get output on paper; speaker is used to receive voice output etc.

SYSTEM UNIT

System unit contains a number of other components which are enclosed in a rectangular casing. The casing is available in two different shapes i.e. vertical shape (called tower casing) and horizontal shape (called desktop casing). The most important component of system unit is a rigid rectangular circuit board called **motherboard**. All other components are etched onto it. It is made up of silicon. An electronic pathway on the motherboard connects different components of system unit with each other. Other components of system unit are RAM, hard disk drive, floppy disk drive, microprocessor etc.

(a) Mouse (b) Keyboard (c) Monitor

(d) System Unit (e) Printer

Figure 2.2: Computer Hardware Components

2.1.2 Computer Software

Computer Software is a term used for organized collections of computer data and instructions. Computer programs are also usually referred to as computer software. A ***Computer Program*** is a set of instructions given to the computer to solve a particular problem. Computer program specify a sequence of operations that computer will perform.

Computer Software is further divided into two major categories

- System Software
- Application software

SYSTEM SOFTWARE

System Software refers to the programs that are responsible for controlling and managing the actual operations of the computer hardware. Generally system software consists of an operating system and some fundamental utilities such as disk formatters, file managers, display managers, user authentication (login) and network control software.

APPLICATION SOFTWARE

Application Software is used to accomplish tasks specified by the user. Application software may consist of a single program, such as an image viewer or a collection of programs (called *software* package) that work together to accomplish a task. e.g. word processor, spreadsheet, database etc.

2.2 Organization of Computer

A computer performsthe following five major functions:

- Accepts data or instructions from input device
- Stores data
- Processes data as required by the user
- Gives results in the form of output
- Controls all operations inside a computer.

In order to perform above mentioned operations; the computer system is divided into three units. These are

- Central processing unit (CPU).
- Memory unit
- Input / Output units

2.2.1 Central Processing Unit (CPU)

The Central Processing Unit (CPU) is generally referred to as the brain of the computer. The primary work of CPU is to convert data from input, process the data

and output in useful information that can be used by either peer or other computer programs. Central processing unit (CPU) is a highly complex set of electronic circuitry that executes program instructions. It is like a really fast calculator that has the ability to recall numbers from various locations in memory, perform arithmetic and logical operations with them, such as addition or multiplication, and then store the results.

All computers must have a central processing unit. As Figure 2.3 shows, the central processing unit consists of two main parts: The control unit, arithmetic and logic unit. Each part has a specific function.

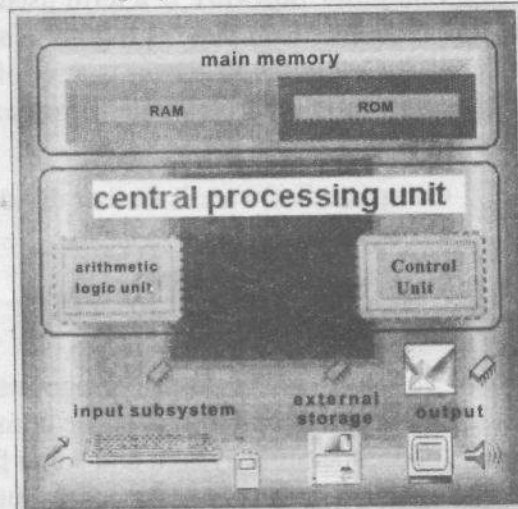



Figure 2.3: Components of Computer System

2.2.2 Arithmetic and Logic Unit (ALU)

The Arithmetic and Logic Unit (ALU) consists of electronic circuitry that executes all arithmetic and logical operations.

The arithmetic and logic unit can perform the following operations, like:

- Addition
- Subtraction
- Multiplication
- Division
- Logical Operations

A logical operation is usually a comparison of numbers, letters, or special characters. The computer can then take action based on the result of the comparison. This is a very important capability. It is by comparing that a computer is able to tell, for example, whether there are available seats on train, whether mobile phone customers have exceeded their pre paid credit limits etc.

Logical operations can test for three conditions:

- **Equal-to condition**: The arithmetic and logic unit compares two values to determine if they are equal. For example, if the number of tickets sold equals the number of seats in the hall, then no more tickets are available.
- **Less-than condition**: The computer compares values to determine if one is less than another. For example, if the hours a person worked this week are less than 35, then some fine is deducted from his salary.

- **Greater-than condition:** The computer determines if one value is greater than another. For example, if the hours a person worked this week are greater than 40, then he gets bonus for working over time.

2.2.3 Control Unit (CU)

The Control Unit consists of circuitry that generates signals to direct the entire computer system to carry out or execute the programs. The control unit itself does not execute program instructions; rather, it directs other parts of the system to do so. The control unit must communicate with the arithmetic and logic unit, memory and other parts of computer system.

It also controls the flow of information through the processor, and coordinates the activities of the other units. This unit also provides clock pulses. Clock pulses are used to regulate and control the speed of all the operations.

In addition to the ALU and CU (Control Unit), the processor has a number of storage locations to store information that is currently being processed. These are called *Registers*. These are temporary storage areas for instructions or data. Registers are managed by the control unit to accept, hold, and transfer instructions or data and perform arithmetic or logical comparisons at high speed.

2.3 System Bus

The CPU must be able to communicate with all devices. The devices are connected together by a communications channel called a **bus**. A bus is composed of a set of communication lines or wires. It is used to move large amount of bits in the form of electrical pulses from a specified source to a specified destination. The bus is used to connect the following units.

- Central Processing Unit
  - * Control Unit
  - * Arithmetic and Logic Unit
- Main memory (RAM , ROM)
- Input / Output Devices

Bus is the common path which is used to send /receive data and commands to / from CPU and memory and all input / output devices. It is also used to send / receive data from secondary storage. The capacity of a bus depends upon the number of data lines it contains. Bus with 16 lines can carry 16-bits (2 Bytes) at a time where as bus with 32 lines can carry 32-bits (4 Bytes)at a time and so on. There are three different buses in the computer system.

- Data Bus
- Address Bus
- Control bus

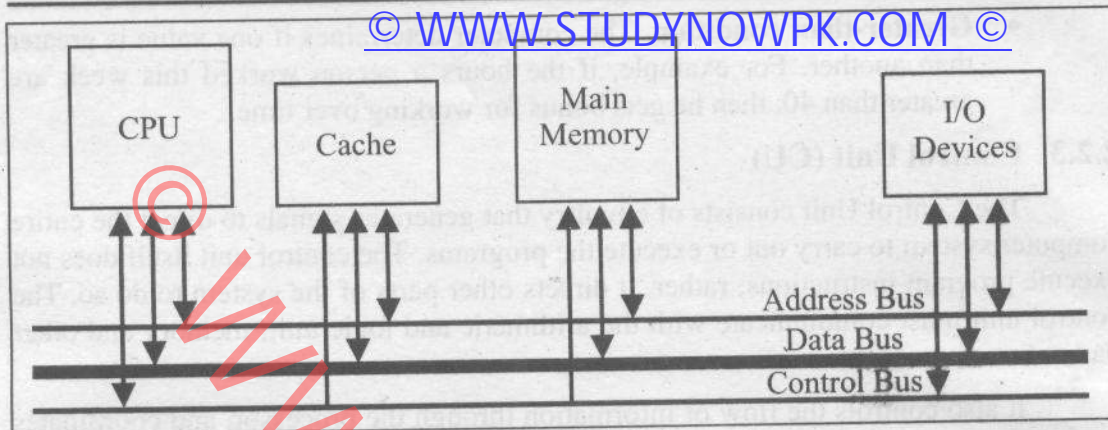



Figure 2.4: System Bus in a typical computer

2.3.1 Data Bus

The most common bus is the data bus. A data bus carries data. It is an electrical path that connects the central processing unit (CPU), memory, input / output devices and secondary storage devices. The bus contains parallel group of lines. The number of lines in the bus affects the speed at which the data travels between different components. Just like the number of lanes on a motorway affect the amount of traffic. Greater number of lanes on road means more cars can pass through it. If more lines are present in the bus, it can carry more data. e.g. a bus with 16 lines can carry 16 bits (2 bytes) and a bus with 32 lines can carry 32 bits (4 bytes) of data.

Busses are designed in such a way that they can communicate with the components in an efficient way. In old computers buses were able to carry only one byte of data but with the improvement in technology the busses present in today's computer can carry multiple bytes at one time thus improving the speed and performance of computers.

2.3.2 Address Bus

An address bus carries address information. It is a set of wires similar to the data bus but it only connects central processing unit (CPU) and memory. Whenever the processor needs data from the memory, it places the address of data on the address bus. The address is carried to the memory where the data from the requested address is fetched and placed on the data bus. The data bus carries it to the processor.

The reason for the importance of address bus is that the number of lines in the address bus determines the maximum number of memory addresses. If an address bus has 8 lines, the maximum number of memory locations that can be addressed are $2^8 = 256$. Today's computers have 32 bit address lines so they can access 4GB (Giga Byte) of memory.

2.3.3 Control Bus

The control bus carries control information from the control unit to the other units. The control information is used for directing the activities of all units. The control unit directs the transfer of data to the ALU from the memory. This data is used by ALU for processing. The control unit also controls the functioning of other units e.g. input/output devices, secondary storage etc.

2.4 Computer Storage

Computer storage is also referred to as **computer memory**. Computer memory is used to store programs and data. There are two types of computer memory.

- Main Storage or Main Memory
- Secondary Storage or Secondary Memory

Main memory is accessible directly by the processing unit. RAM is an example of main memory. As soon as the computer is switched off the contents of the main memory are lost. You can store and retrieve data much faster with main memory as compared to secondary memory. The reason is that the main memory is present on the mother board. Secondary memory such as floppy disks, magnetic disk, etc., is located outside the motherboard. Main memory is more expensive than secondary memory. That is why the size of main memory is less than that of secondary memory.

As the storage capacity of the main memory is limited and often it is necessary to store hundreds of millions of bytes of data for the CPU to process. Therefore, additional memory is required in all the computer systems. This memory is called **secondary memory or secondary storage**.

2.5 Input / Output Devices

A computer is only useful when it is able to communicate with the external environment. When we work with the computer, we feed data and instructions through some devices to the computer. These devices are called ***input devices***. Similarly computer, after processing data and instructions, gives output through some devices called ***output devices***. Input/output devices are also known as *peripheral devices*.

2.6 Ports

A ***port*** can be defined as a socket that enables an external device such as a printer to be attached to the computer". All communication between a computer and external devices is the result of properly connected ports. On every computer, a port's connectors are attached to a motherboard. There are three basic types of ports:

- Serial port
- Parallel port
- USB (Universal Serial Bus) port

Today's computers have all these types of ports with each type doing a different function (See Fig. 2.5, 2.6, 2.7).

2.6.1 Serial port

A serial port allows a serial hardware device to communicate with the computer by transmitting one bit of information at a time. Serial devices, such as mouse, modems, and keyboards, do not require fast data transmission rates. Serial ports often referred to as communications (COM) ports. It is an external port on the back of the computer that attaches directly to the PC's motherboard. These ports were one of the early ports put on computers. Older serial ports also used 25-pin connectors for their serial port, but most new computers with serial ports feature only nine pins.

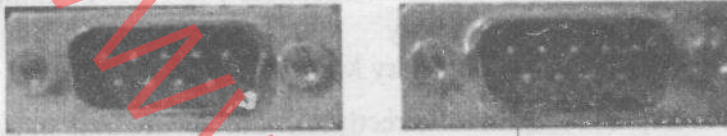



Figure 2.5 : Serial Ports

2.6.2 Parallel ports

A parallel port lets an external parallel device communicate with the computer by transmitting more bits (Such as 8 or 25) of data at a time therefore, it is much faster than a serial port. Most devices that send or receive large amounts of data, such as printers and scanners, use parallel ports. Parallel ports are often referred to as Line Printer (LPT) ports. The parallel port is the largest port on the rear of your PC, comprising 25 lines that include 17 signal lines and eight ground lines.

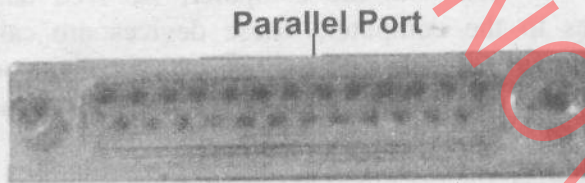



Figure 2.6 : Parallel Port

2.6.3 USB Ports

USB (Universal Serial Bus) is a plug-and-play hardware interface for peripherals such as the keyboard, mouse, joystick, scanner, printer and modem. USB has a maximum bandwidth of 12 Mbits/sec and up to 127 devices can be attached. With USB, a new device can be added to the computer without having to add an

adapter card. It typically is located at the back of the PC and looks as in the following Figure (sometimes positioned vertically). Sometimes it has the USB symbol next to it.

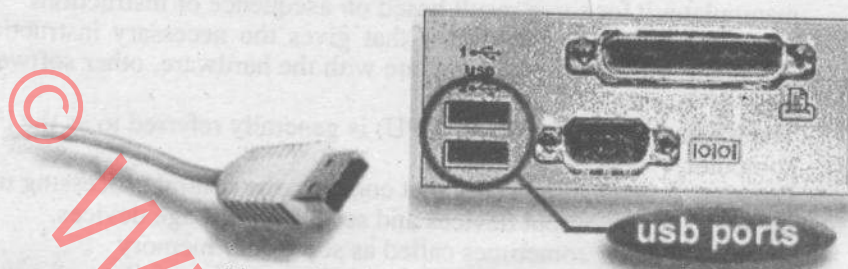



Figure 2.7 : USB port and Connector

Exercise

1. What is the difference between computer hardware and software?
2. Write short notes on the following:
   a. Central Processing Unit
   b. Arithmetic and Logic Unit
   c. Control Unit
3. What is a System Bus?Differentiate between data bus, address bus and control bus?
4. Differentiate between system software and application software?
5. Write a note on main memory and secondary memory? Also give examples.
6. What are the different kinds of input devices?
7. Define magnetic disk.
8. Explain Random access memory.
9. Fill in the blanks:
   (i) The process of performing arithmetic and logic operations with the help of computer is known as ________.
   (ii) Physical parts of the computer system that you can touch and feel are known as ________.
   (iii) Most important part of the computer hardware is the ________.
   (iv) An ________ is a set of programs that gives the necessary instructions to the computer, telling it how to operate with the hardware, other software and the user.
   (v) The ________ and ________ unit contains the electronic circuitry that executes all arithmetic and logical operations .
   (vi) The devices are connected together by a communications channel called a ________.
   (vii) CPU is also called the ________ of the computer .
   (viii) ________ is volatile memory.
   (ix) Magnetic Tape is a ________ device.
   (x) USB stands for ________ .

10. Mark as True or false:

(i) A computer is a device that accepts information in the form of data and manipulates it for some result based on a sequence of instructions

(ii) An RAM is a set of programs that gives the necessary instructions to the computer, telling it how to operate with the hardware, other software and the user.

(iii) The Central Processing Unit (CPU) is generally referred to as the brain of the computer.

(iv) Data bus is an electrical path that connects the central processing unit (CPU), memory, input / output devices and secondary storage devices.

(v) Main Memory is sometimes called as secondary memory

(vi) The memories, which do not loose their content on failure of power supply, are known as **non-volatile** memories.

(vii) A serial port allows a serial hardware device to communicate with the computer by transmitting one bit of information at a time.

(viii) Address bus connects CPU, RAM and other I/O devices.

(ix) Registers are high speed memory locations inside CPU

(x) Instructions in ROM are used in booting process

11. Chose the correct option :

(i) Which of the following devices in not inside the system unit ?
a. Random Access Memory (RAM) **b.** Monitor **c.** The Hard Disk
d. CD-ROM Drives **e.** The Modem

(ii) Which of the following is a part of computer system?
a. Central Processing Unit **b.** Memory **c.** Input / Output Units
d. All of the above **e.** None of the above

(iii) The arithmetic/logic unit can perform:
a. Addition **b.** Subtraction **c.** Multiplication **d.** All of the above
e. None of the above

(iv) The system bus is used to connect the following units.
a. Central Processing Unit **b.** Main memory (RAM , ROM)
c. Input / Output Devices **d.** All of the above **e.** None of the above

(v) Which of the following is not a type of bus in computer ?
a. Data Bus **b.** Address Bus **c.** Power Bus **d.** Control Bus **e.** All of the above

Answers

Q.9

(i) Electronic data processing (EDP) (ii) Computer hardware (iii) Central processing unit (CPU)
(iv) operating system (OS) (v) Arithmetic, Logic (vi) Bus
(vii) Brain (viii) RAM (ix) Storage (x) Universal Serial Bus

Q.10

(i) T	(ii) F	(iii) T	(iv) T	(v) F
(vi) T	(vii) T	(viii) F	(ix) T	(x) T

Q.11

(i) b	(ii) d	(iii) d	(iv) d	(v) c

Input/Output Devices

Input and output devices enable computer to interact with the external world. User can enter data and instructions into the computer with the help of input devices. Whereas computer after processing the data, sends the results back to the output devices. The input can be given in variety of forms e.g. text, image, voice etc. Similarly output may be required in different forms depending upon the user's requirements. That's why a number of different types of input and output devices are available.

3.1 Input Devices

The devices which are used to enter data and instructions into the computer are called input devices.

The most commonly used input devices are:

- Keyboard
- Mouse
- Microphone etc.

Before data processing, data and instructions must be entered into the computer through some input device. The input device converts data and instructions into a form that a computer can process. The computer can process millions of instructions in a second and finally gives output in the form of information. The figure 3.1 shows the different categories of input devices.

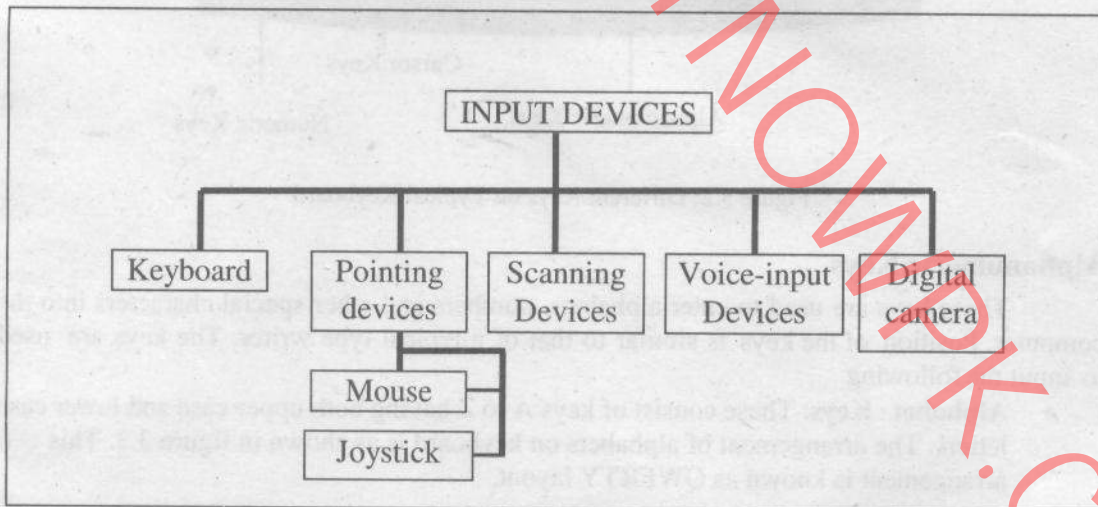



Figure 3.1: Categories of input devices

3.1.1 Keyboard

Keyboard is the standard input device used to enter textual data into the computer. The layout of keyboard is just like the traditional typewriter. But it contains some extra command keys and function keys. Among the most popular keyboard layout are QWERTY and DVorak. The layout refers to the arrangement of keys on the keyboard. A typical keyboard can have 101 to 104 keys.

The keys on computer keyboards are often classified as follows:

- Alphanumeric keys: Alphabets and numbers.
- Numeric keys: Numbers and mathematical operators.
- Function keys: keys for special functions (F1, F2, F3 ….. F12).
- Cursor control keys: Keys that move the cursor (Up, Down, Left, Right).

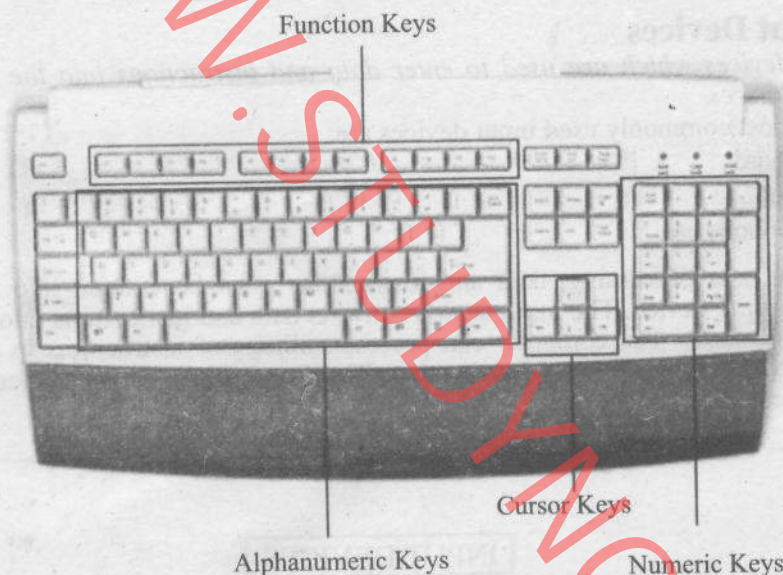



Figure 3.2: Different Keys on Typical Keyboard

Alphanumeric Keys

These keys are used to enter alphabets, numbers and other special characters into the computer. Position of the keys is similar to that of a typical type writer. The keys are used to input the following

- **Alphabet Keys**: These consist of keys A to Z having both upper case and lower case letters. The arrangement of alphabets on keyboard is as shown in figure 3.3. This arrangement is known as QWERTY layout.

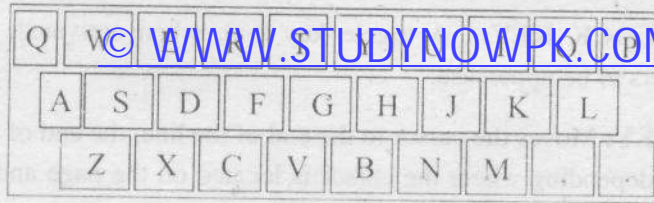



Figure 3.3 : Alphabets Layout on keyboard (QWERTY Layout)

- **Number keys:** These consist of number keys starting from 0 to 9 .
- **Special Characters keys:** These consist of punctuation keys, special characters, and space bar.

Numeric Keys

These keys are used to input numbers into the computer. The layout of these keys is similar to that of a typewriter. Alongwith the number keys it also contains keys for mathematical operations i.e. plus, minus, multiply, divide.

Function Keys

Function keys are used to perform different functions depending upon the application or program that is executing. These keys provide shortcuts for doing routine tasks on a computer.

Most computer keyboards have a row of Function keys at the top of the keyboard. These keys are marked from F1 to F12. Many programs, including most of Microsoft's products, support use of the function keys. The function keys are frequently used in combination with other keys such as the CTRL key, the ALT key, and the SHIFT key. This results in a large number of possible keyboard shortcuts.

Cursor Control Keys

Cursor is used to represent the position where the keyboard's input will be placed. These four arrow keys are used for moving the cursor from its current position to *right, left, up,* or *down*. Moving these cursor keys does not change or delete any characters on the screen. These are also used for screen navigation. The function of some of the other important keys is given below:

ENTER KEY: Used to enter commands or to move the cursor to the beginning of the next line. This is sometimes labeled as *Return* instead of Enter.

ESC KEY: Short for Escape, this key is used to send special codes to devices and to exit (or escape) from programs and tasks.

DELETE KEY: Deletes the character at the current cursor position and the one at right of the cursor position. Also used to delete the selected object, but it does not move the cursor.

CAPS LOCK KEY: A toggle key that, when activated, causes all alphabetic characters to be uppercase.

END KEY: Moves the cursor to the end of the line, the end of the page, or the end of the file depending where the cursor is located on the page and on which program is running.

CTRL KEY: Short for Control, this key is used in combination with other keys to produce special characters. The meaning of each control character depends on which program is running.

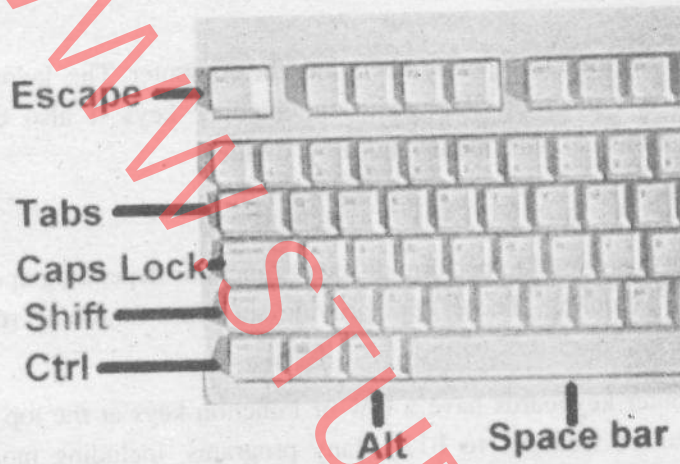



Figure 3.4 : Special Keys on Left Side of keyboard

ALT KEY: Short for Alternate, this key is used in combination with other keys to produce special characters.

TAB: This key enables the cursor to jump a couple of spaces to the right on the screen. To jump the equivalent spaces to the left, Shift-Tab keys should be pressed together.

BACKSPACE KEY: Deletes the character present at the left of the cursor and moves the cursor to that position.

PAGE UP and **PAGE DOWN:** This is used to move the cursor up or down a certain fixed number of lines usually one page at a time. This is often abbreviated as PgUp and PgDn.

HOME: Moves the cursor to the left corner of the line or to the beginning of the page or file depending where the cursor is and on which program is running.

INSERT: In insert mode, all characters typed are placed at the cursor position (or to the right of the insertion point). With each new insertion, characters to the right of the

cursor are pushed to the right of the insertion point to make room for the new characters. If insert mode is turned off, typing then overwrites existing characters instead of inserting the new ones before the old ones. This is often called overwrite mode. Most PC keyboards have an INSERT key that lets you switch back and forth between insert and overwrite mode.

Space Bar: This key moves the cursor one space to the right every time you press key.

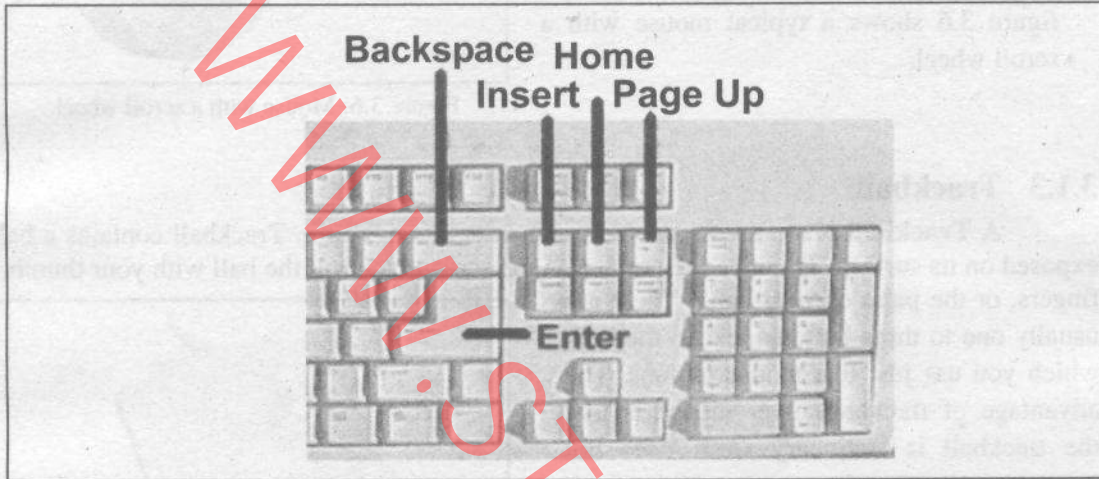



Figure 3.5: Special Keys on right side of keyboard

3.1.2 Mouse

A mouse is an input device that rolls around on a flat surface and controls the pointer on a display screen. The pointer is an on-screen object – usually an arrow – that is used to select text; access menus; and interact with programs, files or data that appear on screen. As you move the mouse, the pointer on the display screen moves in the same direction. A typical mouse has two buttons, which perform different functions depending on what application is running. Some mouses also include a scroll wheel for scrolling through long documents.

Optical mouse is getting popular nowadays; instead of having a roller ball it uses light reflection to control the movement of pointer on the screen.

Mouse Events:

Mouse event refers to the activity that can be performed by using the mouse. A typical mouse can perform the following events:

- Left Click
- Right Click
- Drag

Left Click is used to select a graphical object such as a file icon and a piece of text in a document etc., or to press a button such as the start button and closing, opening or minimizing a window etc.

Right Click is used to view the properties of an object such as file, folder, desktop etc.

Drag event triggers when you keep on pressing the left mouse button while moving the mouse. This event is used to select more than one items at a time, or to drop an object into another application. The figure 3.6 shows a typical mouse with a scroll wheel.

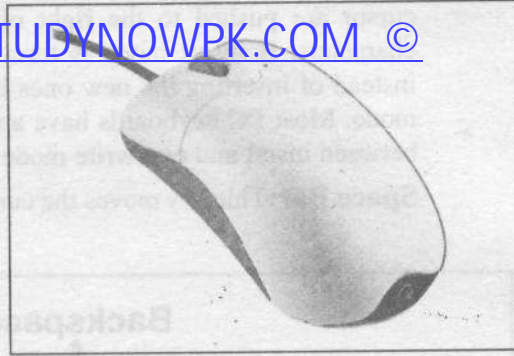



Figure 3.6: Mouse with a scroll wheel

3.1.3 Trackball

A **Trackball** is a pointing device that works like a mouse. Trackball contains a ball exposed on its surface. To move the pointer on the screen, you roll the ball with your thumb, fingers, or the palm of your hand. There are usually one to three buttons next to the ball, which you use just like mouse buttons. The advantage of trackballs over mouse is that the trackball is stationary so it does not require much space for use. In addition, you can place a trackball on any type of surface. For both these reasons, trackballs are popular pointing devices for portable computers.

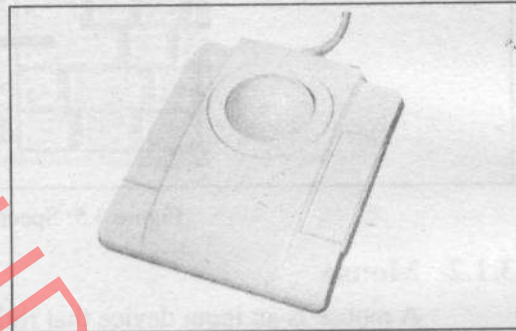



Figure 3.7: A Typicall Trackball

3.1.4 Joystick

Joystick is an input device used for games, computer aided designs or simulations. It is like a lever that moves in all directions and controls the movement of a pointer or some other display symbol. A joystick is similar to a mouse, except that with a mouse the cursor stops moving as soon as you stop moving the mouse. With a joystick, the pointer continues moving in the direction the joystick is pointing. To stop the pointer, you must return the joystick to its neutral position. Most joysticks include two buttons called triggers.

Figure 3.8: Joystick

3.1.5 Scanner

A **scanner** is an input device which is capable of reading the image present on a piece of paper and transferring the information into a computer where a program can store and interpret it. This information is in the form of a graphic image or picture and not in the form of text. Even if the paper contains text, the information transferred is not text but a picture of text. Optical Character Recognition (OCR) software is necessary to read this image and convert it to actual text data and this software varies widely in its ability to do so. Scanners range from monochrome (2 color) scanners to flatbed scanners and color scanners capable of scanning a whole page at once and costing several thousand rupees.

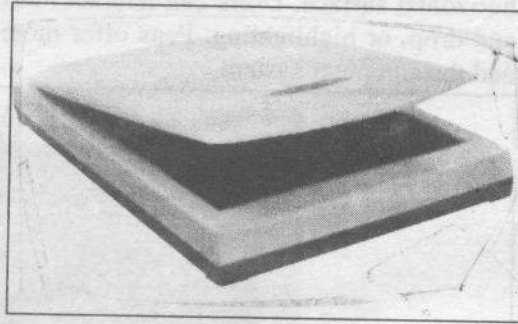



Figure3.9: Scanner with open top cover

3.1.6 Microphone

A **microphone** is an input device used to digitally record audio data, such as the human voice. It can be plugged into a computer or recorder. Many productivity applications can accept input via a microphone, enabling the user to dictate text or issue commands orally. Software in the computer converts the sound impulses into digital form. It is then stored in memory and processed when required.

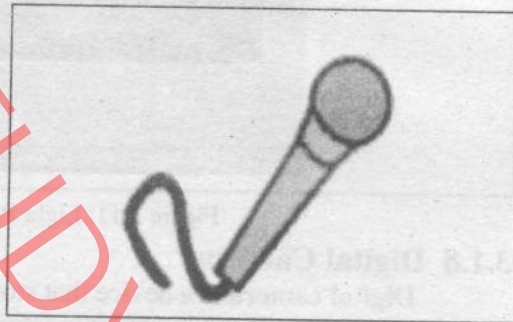



Figure 3.10: Typical Microphone

Voice Recognition

A **voice-recognition** system, using a microphone as an input device, converts a person's speech into digital data by comparing the electrical patterns produced by the speaker's voice with a set of prerecorded patterns stored in the computer.

Voice Recognition allows a user to use his voice as input. Voice recognition may be used to dictate text to the computer or to give commands to the computer such as opening application programs, pulling down menus, or saving work.

Older voice recognition applications require each word to be separated by a distinct pause. This allows the machine to determine where one word begins and the next stops. This style of dictation is called *discrete speech*. Many people prefer these systems to the newer continuous speech.

3.1.7 Light Pen

Light pen is a light-sensitive input device shaped like a pen. It is used to draw on the computer screen or to make menu selection. As the tip of the light pen makes contact with the

screen, it sends a signal back to the computer containing the x-y coordinates of the point. Light pens can be used on any size screen.

Light pens give the user the full range of mouse capabilities, without the use of any horizontal surface. Users can interact more with the application, using options such as drag and drop, or highlighting. Pens offer more active and accurate interaction between the user and the computer system.

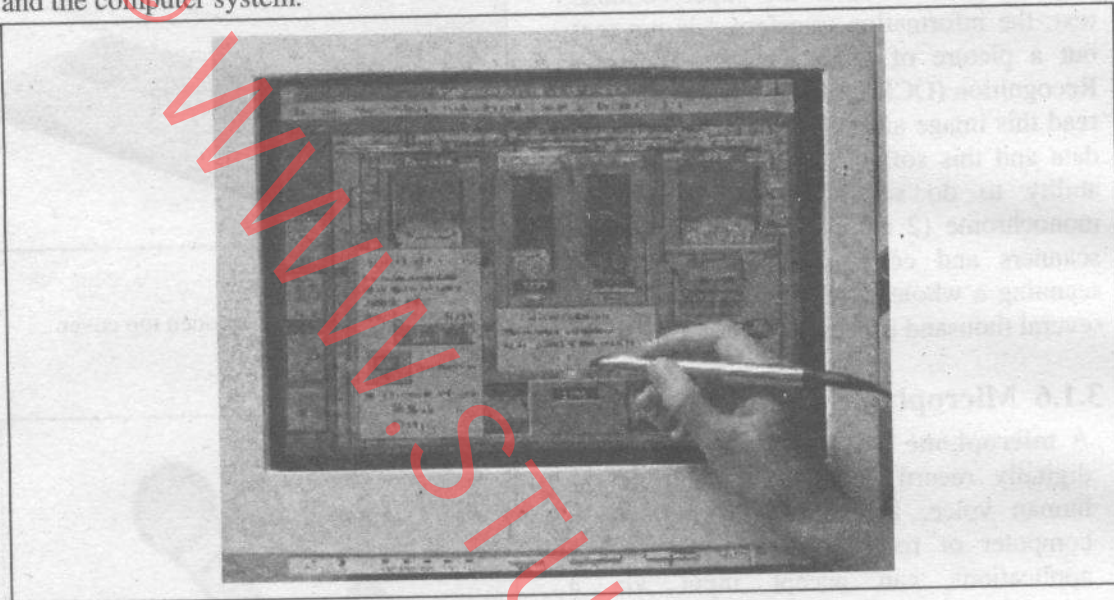



Figure 3.11: Light Pen used on a graphic display

3.1.8 Digital Camera

Digital camera is a device that stores images digitally in its memory rather recording them on film. Once a picture has been taken, it can be downloaded to a computer system, and then manipulated with a graphics program and printed. Digital photos are limited by the amount of memory in the camera and by the quality of the final output device.

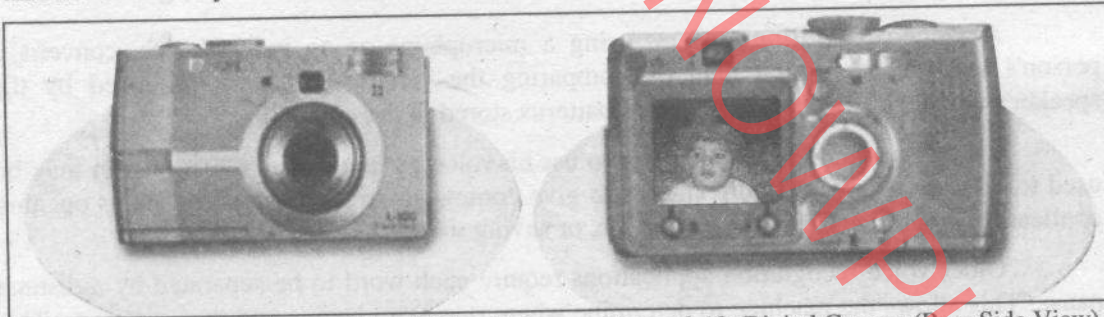



Figure 3.12:Digital Camera (Front Side view) Figure 3.13: Digital Camera (Rear Side View)

The big advantage of digital cameras is that making photos is both inexpensive and fast because there is no film processing involved.

3.1.9 Disk Drive

A **disk drive** is a machine that reads data from and writes data onto a disk. Disk drive rotates the disk with precise timing and has one or more read/write heads that read and write

data. The data is read by the read/write head and transferred to the computer's main memory for further processing. There are different types of disk drives for different types of disks e.g., a *hard disk drive* (HDD) reads and writes hard disks, and a *floppy disk drive* (FDD) accesses floppy disks, and an *optical disk drive* reads optical disk. Disk drives can be either *internal* (housed within the computer) or *external* (housed in a separate box that connects to the computer).

3.2 Output Devices

*The devices which are used to receive data and information from the computer are called **output devices**.*

The most commonly used output devices are:

- Monitor
- Printer
- Speaker etc.

Before going into details of output devices it is necessary to discuss two basic concepts related to input/output devices i.e. hard copy and soft copy. An electronic version of a document stored (normally as a file) on a storage device is called **soft copy,** whereas the printed version of an electronic document (on a paper) is called **hard copy**. Here is a brief discussion on commonly used output devices.

> **Remember that:**
> There are some devices which act as both input and output devices e.g. touch screen, hard disk drive, floppy disk drive etc.

3.2.1 Monitor

The **monitor** is the most commonly used output device on personal computers (PCs). It is also called a display or a screen. You look at your monitor whether you are typing, issuing commands, surfing the Internet, or even listening to music. The quality of the image on the screen depends upon many factors. **Resolution** is one of the factors affecting the quality of the monitor. The *resolution* of a monitor refers to the number of pixels on the screen, expressed as a matrix.

All the monitors can be categorized by the way they display colors. These are monochrome monitors and color monitors. *Monochrome monitors* can display only one color (such as green, amber or white) against a contrasting background, which is usually black. These monitors can display text only and are not used for displaying graphics. *Color monitors* display combinations of red, green and blue colors. These three are the basic colors and their combination can display a full range of colors.

TYPES OF MONITOR

There are two basic types of monitors. These are

- CRT (cathode rays tube) monitor
- Flat panel monitor

CRT MONITOR

A CRT monitor consists of a phosphorus coated screen; there are three electron guns on its back. The screen's phosphor coating is organized into a grid of dots. The smallest number of phosphor dots that the gun can focus on is called a **pixel** or **picture element**.

The three electron guns emit beams of three different colors i.e. red, green and blue. In color monitors each pixel includes three phosphors – red, green, and blue arranged in a triangle. When the beams of each of these guns are combined and focused on a pixel, the phosphors lights up. The monitor can display different colors by combining various intensities of the three beams.

A CRT monitor contains a **shadow mask**, which is a fine mesh made up of metal, fitted to the shape and size of the screen. The holes in the shadow mask's mesh are used to align the electron beams, to ensure that they strike precisely the correct phosphor dot. In most shadow masks, these holes are arranged in triangle.

Figure 3.14: A CRT Monitor

FLAT-PANEL MONITOR

Flat panel monitors are usually used in portable computers and laptops because of their small size. They are expensive to manufacture and don't provide the high quality and bright colors that CRT technology provides. There are several types of flat-panel monitors but the most common is liquid crystal display (LCD) monitor.

LCD monitors create images with a special kind of liquid crystal that is normally transparent but becomes opaque when charged with electricity. If you have a handheld calculator or a digital watch, it probably uses a liquid crystal display.

Figure 3.15: Typical LCD Screen

VIDEO CONTROLLER

The quality of the images that a monitor can display is defined as much by the **video controller** as by the monitor itself. Video controller is an intermediary device between the CPU and the monitor. It contains memory and circuitry necessary to send information to the monitor for display on screen. The resolution of the monitor is actually determined by the video controller, not by the monitor itself e.g. Video Graphic Array (VGA) has resolution 640 x 480 pixels, Super Video Graphic Array (SVGA) has resolution 1024 x 768 etc.

3.2.2 Printer

A *printer* is a device that produces hardcopy on the paper. Printers are commonly used in businesses to get the documents printed on paper. Depending upon the printing mechanism used, printers fall into two categories i.e.

- Impact printers
- Non impact printers

IMPACT PRINTERS

An **impact printer** produces images by striking an inked ribbon with a hammer or a set of pins, pressing ink from the ribbon onto a piece of paper.

Impact printers are the oldest print technologies which are still in production. Some of the largest printer companies continue to manufacture, sell, and support impact printers. Impact printers are most useful in specialized environments where low-cost printing is required. The three most common forms of impact printers are:

- Dot-Matrix Printer
- Daisy-Wheel Printer
- Line Printer

Dot-Matrix Printer produces characters by striking pins against an ink ribbon to print closely spaced dots in the appropriate shape. The shape forms a number, alphabet or other special characters. Dot-matrix printers are relatively expensive and do not produce high-quality output; however, they can print multiple copies of a page at a time.

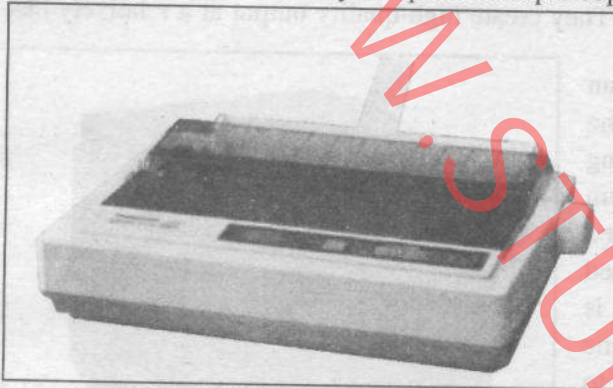



Figure 3.16: Dot Matrix Printer

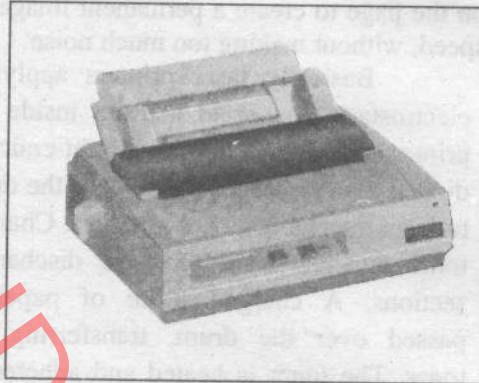



Figure 3.17: Daisy Wheel Printer

Daisy-wheel Printer has characters etched at the outer edge of a pedaled wheel (hence the name daisy wheel) which forms characters in the same way as a typewriter. Daisy-wheel printers are loud and slow. They cannot print graphics, and cannot change fonts unless the print wheel is physically replaced. With the advent of laser printers, daisy-wheel printers are generally not used in modern computing environments.

Line Printer is somewhat similar to the daisy-wheel is the line printer. Line printers have a mechanism that allows multiple characters to be simultaneously printed on the same line. Line printer has speeds ranging from 300 LPM (lines per minute) to 2400 LPM.

Because of the nature of the print mechanism, line printers are much faster than dot-matrix or daisy-wheel printers; however, they tend to be quite loud, have limited font capability, and often produce lower print quality than recent printing technologies.

Because line printers are used for their speed, they use special paper with pre-punched holes along each side. This arrangement helps to continue printing at high speed till the paper runs out.

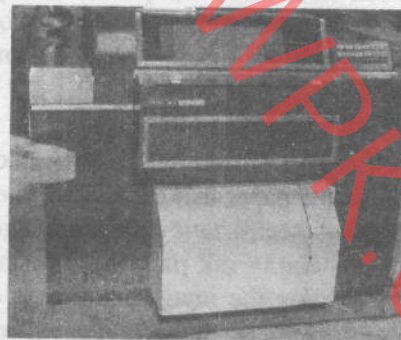



Figure 3.18: Line Printer

NON-IMPACT PRINTER

A **non-impact printer** produces images on paper without striking the page in any way.

There are several types of non-impact printers called thermal and electrostatic printers. These printers use a chemically coated paper on which the characters are exposed by some means such as a laser. This means that these printers can produce a printed image without striking the paper. Because the printing device is simple and has no moving parts, these printers are inexpensive to manufacture and silent. Very fast non-impact printers are capable of printing more than 24 pages per minute. Different kinds of non-impact printers are explained below.

LASER PRINTER

Laser stands for Light Amplification by Stimulated Emission of Radiations. Laser printers are similar to copy machines. They use laser beams to burn special ink called toner on the page to create a permanent image. They create high-quality output at a relatively fast speed, without making too much noise.

Basically laser printers apply an electrostatic charge to a drum inside the printer cartridge. A laser or a light-emitting diode then discharges portions of the drum to form the characters or graphics. Charged toner attaches itself to these discharged sections. A charged piece of paper is passed over the drum, transferring the toner. The toner is heated and adhered to the sheet.

Laser printers have speeds of 4, 8, 12, or more pages per minute. These speeds can vary depending on whether you are printing plain text, or images.

Figure 3.19: Laser Printer

ELECTRO-THERMAL PRINTER

An electro-thermal printer is a type of printer that uses heated pins to "burn" images onto heat-sensitive paper. These types of printers are commonly used in calculators and fax machines. They are inexpensive and print relatively fast but produce low resolution prints.

ELECTROSTATIC PRINTER

Electrostatic printers use a special photographic paper that allows characters to be etched onto the paper using a stylus. The stylus is made up of tiny wires and forms characters by placing an electrostatic charged image on the paper. Then, as the paper is moved through a solution containing ink particles, the ink adheres to the charges that form a pattern on the paper to develop the image. This type of printer can be used for both printing and plotting (displaying graphic output), and can print more than 5,000 lines per minute.

INK JET PRINTER

Ink-jet printer is a type of printer that works by spraying ink on a sheet of paper. Magnetized plates in the path direct the ink onto the paper in the desired shapes. Ink-jet printers are Capable of producing high quality print that are produced by laser printers. A typical ink-jet printer provides a resolution of 300 dots per inch, although some newer models offer higher resolutions. These printers can also produce high quality color graphics including photos.

Figure 3.20: Ink Jet Printer

In general, the price of ink-jet printers is lower than that of laser printers. However, they are considerably slower. Another drawback of ink-jet printers is that they require a special type of ink

3.2.3 Plotter

A **plotter** is a large scale printer that receives commands from a computer to make drawings on the paper with one or more automatic pens. Unlike a regular printer, the plotter can draw continuous point-to-point lines directly from computer graphics files or commands. There are three basic types of plotters, i.e,

- Drum plotters
- Flatbed plotters
- Electrostatic plotters

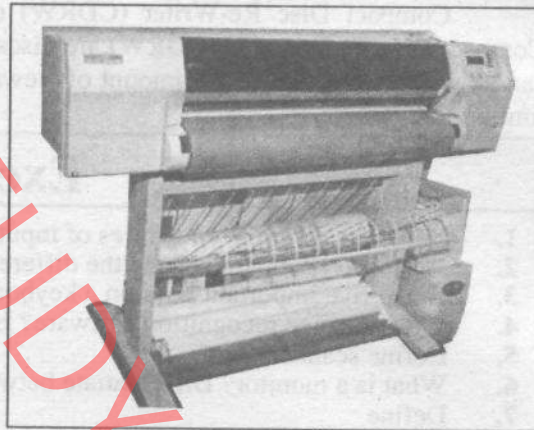



Figure 3.21: Plotter

DRUM PLOTTER

The printing mechanism of the **drum plotter** involves a pen and a drum. The paper is wrapped onto the drum that rotates back and forth. To produce an image onto the paper, the pen (mounted on a cartridge) moves horizontally while the rotation of drum causes the paper to move vertically. In this way the vertical movement of the paper and the horizontal movement of the pen create the required design. Pen having different colors can be used to produce output in different colors.

FLATBED PLOTTER

The printing mechanism of **flatbed plotters** consists of two arms and a rectangular flatbed. Flatbed plotters use two arms, each of which holds a set of colored ink pens. The two arms operate at right angle as they draw on a stationary piece of paper. Flatbed plotters are very slow and even can take hours to print a complicated drawing.

ELECTROSTATIC PLOTTER

Electrostatic Plotter draws on negatively charged paper with positively charged toner. As a rule, plotters are much more expensive than printers. They are most frequently used for CAE (Computer-Aided Engineering) applications, such as CAD (Computer-Aided Design) and CAM (Computer-Aided Manufacturing).

COMPACT DISK-RECORDABLE/ RE-WRITABLE (CD-R/CD-RW)

A CD-Recordable (CD Writer) is a drive that allows you to record information to a CDR disc (Compact Disk Recordable). The recorded information can be a mixture of data, digital audio, and video. A CDR disc will hold a maximum of 700 megabytes of data or 80 minutes of digital audio or video. In cases of mixing formats, the amount of information is proportional to each other.

Figure 3.22: Compact Disc Re-Writer (CDRW) drives

Therefore, 350 megabytes of data used on a disc means that you have 40 minutes of digital audio or video left. CDR discs can be used on almost any CD-ROM drive, CD Audio player, and CD Video player. A CDR disc can be written to only once and can not be deleted once it has been recorded. There are ways of writing information in parts, also known as multi-session.

Compact Disc Re-Writer (CDRW) drives can use both CDR and CDRW discs. Compact Disc Re-Writable (CDRW) are discs on which old data can be erased and new data can be written. Normally, the amount of "rewrites" that you can do on a CDRW disc is 1000 times.

Exercise

1. Name the different categories of input devices.
2. What is a keyboard? Name the different key categories on the keyboard.
3. Name five important keys on a keyboard along with their functions.
4. What is voice recognition software? Explain.
5. Define scanners.
6. What is a monitor? Differentiate between monochrome and color monitors.
7. Define
   a. Flat Panel Display
   b. Liquid Panel Display
8. Write a note on different kinds of printers.
9. What is a plotter?
10. What is the difference between soft copy and hard copy?
11. Fill in the blanks:
    (i) A ____________ is the most commonly used input device that enables you to enter data into a computer.
    (ii) ____________ keys are used to enter alphabets, numbers and other special characters into the computer.
    (iii) ____________ keys are used to input numbers into the computer.
    (iv) ____________ is an optical disc used for storing digital data.
    (v) ____________ used to store large quantities of data inexpensively and therefore are often used for backup.
    (vi) QWERTY is a ____________ .
    (vii) Color monitor uses ____________, ____________ and ____________ colors to display a colored picture.
    (viii) Scanner is a(n) ____________ .
    (ix) LASER stands for ____________ .
    (x) CD can store more than ____________ data.

12. Mark as true or false .

(i) Mouse allows the user to control the movement of the cursor or pointer on a display screen.

(ii) Joystick is an input device used for games, computer aided designs or flight simulators.

(iii) A scanner is an input device which is capable of reading the image present on a piece of paper and transferring the information into a computer where a program can store and interpret it.

(iv) Trackball is an output device.

(v) Monochrome monitors can display all the colors.

(vi) Touch Screen is an input as well as output device.

(vii) Hard disk is a sequential access device.

(viii) Megnatic tape is normally used to take backup of data .

(ix) Impact printers don't touch the surface of paper during printer .

(x) Mouse is the primary (inputdevice) .

13. Chose the correct answer.

(i) Which of the following is not an input device?

a. Magnetic tape units **b.** Floppy disk drive units
c. Monitor **d.** Keyboard **e.** Mouse

(ii) The keys on computer keyboards are often classified as follows:

a. Alphanumeric keys **b.** Numeric keys **c.** Function keys
d. Cursor control keys **e.** All of the above

(iii) Which printer uses laser beams to burn special ink called toner on the page to create a permanent character?

a. Dot matrix **b.** Daisy wheel **c.** Laser **d.** Ink Jet **e.** Plotter

(vi) Flat panel displays are usually used in

a. Supercomputer **b.** Personal Computers **c.** Portable Computers / Laptops
d. All of the above **e.** None of the above

(v) Hard disk is a ________ disk that stores data.

a. Optical **b.** Magnetic **c.** Random Access **d.** Read only **f.** None of the above

Answers

Q.11

(i) Keyboard (ii) Alphanumeric keys (iii) Numeric keys (iv) CD-ROM
(v) Magnetic Tape (vi) Keyboard layout (vii) Red, Green, Blue (viii) Input
(ix) Light Amplification by stimulated emission of radiations (x) 700 MB

Q.12

(i) T	(ii) T	(iii) T	(iv) F	(v) F
(vi) T	(vii) F	(viii) T	(ix) T	(x) F

Q.13

(i) c	(ii) e	(iii) c	(iv) c	(v) b

SRP

Chapter 4

Storage Devices

Computers are used to process large volumes of data and to execute very complex programs. The computers need to have some kind of storage device to hold these programs and data. Such a device should be directly accessible to the CPU and its speed must be compatible with the speed of the CPU. Also a computer must be able to store frequently needed data on some permanent storage device. Based on the characteristics of the storage devices we classify these devices as Main Memory or secondary storage device. In this chapter we will study the need and use of basic storage devices used with the computers.

4.1 Main Memory

Digital computers are stored-program computers that means a program to be executed is first loaded in the memory and then instructions are executed one by one. The data and results of calculations are also stored in the memory. In this sense main memory is the working area of the computer. It is very fast but limited in capacity. A computer cannot work without having some kind of main memory. Most general purpose computers have enough memory to store a few million characters. In this section we will learn about types of main memory, their use and working principles.

The main memory of a computer consists of thousands or even millions of cells, each capable of storing a bit i.e, 0 or 1. These cells are logically organized into group of 8 bits (Binary digits) called a byte as shown in the Figure 4.1.

Figure 4.1: Memory cells organized as a byte

Each byte in the memory has a unique number assigned to it. This number is called the address of that byte. This scheme of arranging cells into a byte and bytes into memory chip is shown in Figure 4.2. The number shows the byte number assigned to the byte and is also called its address.

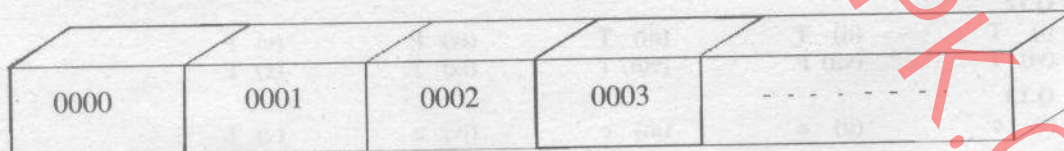



Figure 4.2: Memory addresses

We can view the main memory of a computer as a collection of bytes arranged in an order or sequence. CPU or any other component of the computer can access any byte from the main memory by specifying its address. Different bytes of the main memory can be accessed directly in a random order in equal amount of time. Because of this characteristic of the main memory it is called **direct access storage device**. Accessing any byte of the memory is very fast as compared to other storage devices like the magnetic and optical disks. Most computers have the following two types of main memory.

4.1.1 RAM (Random Access Memory)

RAM is the primary storage device and the data and instructions are stored temporarily in it. It takes the same amount of time to access any location in RAM. CPU can perform two types of operations on RAM, these are:

- Read
- Write

During **Read operation** the contents of memory location are copied to a CPU register whereas during **Write operation** the contents of a CPU register are copied to the memory location. The CPU can not perform any other operations on memory locations. RAM is usually built by using two different technologies i.e. DRAM (Dynamic RAM) and SRAM (Static RAM).

DRAM is the most commonly used technology to build RAM chips and consumes a lot of power as data stored in a DRAM needs to be refreshed periodically.

SRAM is faster than the DRAM but it is more expensive. Unlike DRAM, the contents of SRAM do not need to be refreshed periodically.

In most computers SRAM technology is used to build very fast memory inside a CPU chip. This memory is known as the **cache memory**. Cache memory usually is very small in size as compared to the total memory in the computer but it increases the performance of a computer. This memory arrangement is shown in the Figure 4.3. Following are the main characteristics of RAM :

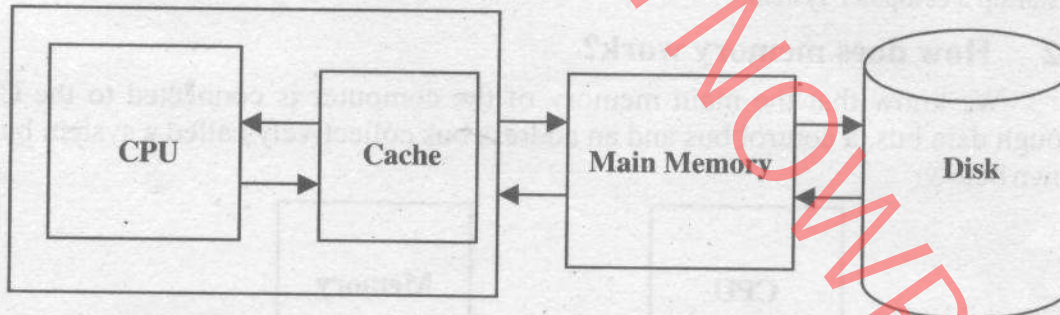



Figure 4.3: Memory management

- The contents of the memory are lost when the electricity supply is cut-off so the main memory is **volatile**.
- Since CPU can read data from and write data to the RAM therefore RAM is read/write memory.
- RAM is random access in the sense that any part of RAM can be accessed directly.

4.1.2 ROM (Read Only Memory)

As is obvious from the name the contents of ROM can be read but new data can not be written into it so it is a **R**ead **O**nly **M**emory. The manufacturer of the ROM writes the data and programs permanently into it and this data and programs can not be changed afterwards. ROM is used to save frequently used instructions and data. The data stored in ROM will not change for a very long time. Following are the commonly used forms of ROM.

4.1.3 PROM (Programmable Read Only Memory)

This form of ROM is initially blank and the user can write his own data/programs on it by using special devices. Once the program/data is written on PROM it cannot be changed or altered. It is obvious that this kind of ROM will be used for storing data for a very long period of time. The data written on this kind of ROM can not be changed once it is written.

4.1.4 EPROM (Erasable Programmable Read Only Memory)

Like PROM it is initially blank and programs and data can be written on it by the manufacturer or by the user with special devices. Unlike PROM a user by using special purpose devices and ultraviolet rays can erase the data written on it. So data/program written on it can be changed and new data can also be added on this form of ROM. As the data written on this kind of ROM can be changed so data that may need to be updated can be written on it but frequently changing data is not written on EPROM.

4.1.5 EEPROM (Electrically Erasable Programmable Read Only Memory)

This kind of ROM can be re-written by using electrical devices and so data stored on the EEPROM can be easily modified. EEPROM can be very useful for taking backup of data and for keeping records that are updated periodically.

It is important to note that all the forms of ROM described above are non-volatile so the data stored in these chips is not lost when electricity is cut-off. Mostly ROM chips are used to store frequently used programs like operating system routines (small programs) and data, which is not changed for long periods of time. It is also used to store programs needed to startup a computer system.

4.2 How does memory work?

We know that the main memory of the computer is connected to the CPU through data bus, a control bus and an address bus collectively called a system bus is shown below:

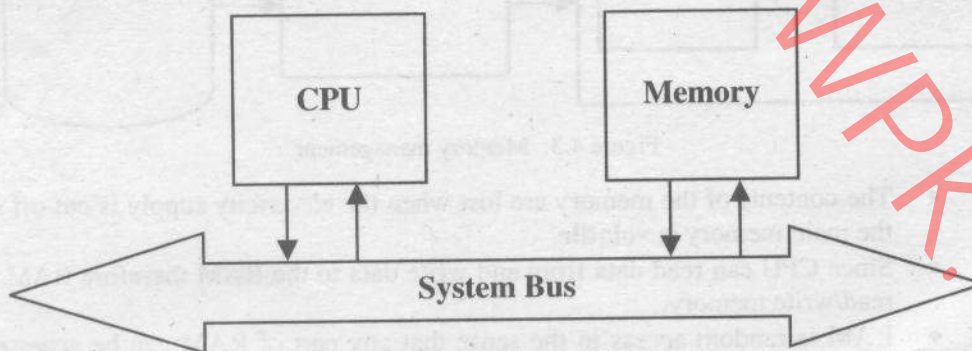



Figure 4.4 : System Bus

When the CPU wants to read data from the memory it first places the read request on the control bus, and also places the address of the byte or word needed on the address bus. The memory unit reads the command and the address and puts the required data on the data bus. The CPU then reads this data from the data bus. Similarly for writing data, CPU first places the Write request on the control bus and also places the address of the word where it wants to write on the Address bus. When memory unit gets ready to do the operation the CPU puts the data on the Data bus and memory unit reads this data and places it in the required word.

As the main memory consists of electronic circuits so a word or byte address is accessible without using any mechanical components. Because of this property the access speed of memory is very fast. Also the data stored in a computer's main memory can be processed in any order. Because of this property the main memory is often referred to as Random Access Memory (RAM). As the RAM is constructed from integrated circuits so it needs to have continuous electrical power supply in order to maintain. When power is switched off all the data stored into it is lost so we say that RAM is volatile.

4.3 Memory Measuring Units

In digital computers the data is represented as a collection of bits. A **bit** is the smallest unit of data that can be used by a computer. We also know that this data is grouped into bytes and a byte is the number of bits needed to store a character. A **byte** is comprised of eight bits. The size of a computer's main memory is often measured as the number of bytes in it. Following is a list of different memory measuring units:

1 Nibble	**= 4 bits**	
1 Byte	**= 8 bits**	
1 KB (Kilo Byte)	**= 1024 bytes**	**= 2^{10} bytes**
1 MB (Mega Byte)	**= 1024 KB**	**= 2^{20} bytes**
1 GB (Giga Byte)	**= 1024 MB**	**= 2^{30} bytes**
1 Terabyte	**= 1024 GB**	**= 2^{40} bytes**

4.4 Data Organization Within a byte or Word

We view the bits within a byte or word as being arranged in a row from left to write. We call one end of this row the high-order end and the other the low-order end. The bit at left end is often called the high-order bit or the **Most Significant Bit (MSB)**;similarly,the bit at the right end is referred to as the low-order bit or the **Least Significant Bit1 (LSB)**.This is shown in Figure 4.5

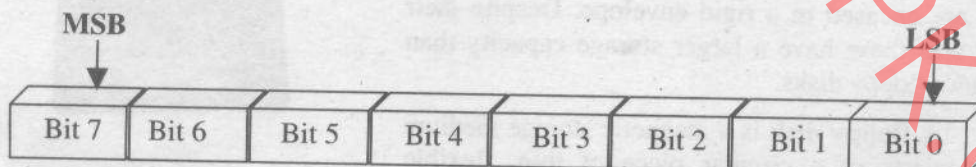



Figure 4.5: A byte with the most significant and least significant bits

4.5 Secondary Memory

Primary memory is directly accessible to the processor and is used to store data and programs that are in current use. The control unit does not have direct access to data that is stored anywhere outside the processor or main memory. However, this storage is limited in size and volatile. We need some storage device that is not temporary in nature and that does not have the same restrictions of size as that of main memory. Such a device is called secondary storage device. Secondary storage devices are categorized according to

- means by which the data is stored, optically or magnetically
- the technique used for storage of the data, sequential storage or direct access storage
- the capacity of the medium, how much can be stored on it
- portability of the medium, can it be moved around easily
- access times to the data stored.

Secondary storage is required to permanently store information that is not needed in memory all of the time and which may be too large to fit into the memory of the computer. Two main categories, based on the ways of accessing data from a secondary storage device, are: **Sequential-Access** and **Direct-Access** or **Serial-Access** and **Random-Access** respectively.

Different computer application programs need these two types of storage devices. For example, a program for calculating the payroll of a company has to access all the data on all of a company's employees, it accesses this data one record at a time, one after the other, and this is called sequential access. Direct access storage device can be used in a departmental store where details of all of the items for sale are needed in a random order. Following table shows a comparison of main memory and secondary memory.

Primary memory	Secondary memory
Expensive	Cheap
small capacity	Large capacity
Connects directly to the processor	Not connected directly to the processor
Fast Access	Slow Access

4.5.1 Floppy disk

Floppy disks are mostly used for transferring data between computer systems and for casual backup of data. They have low capacity, and are very, very slow as compared to other storage devices. The most common size is 3.5 inches diameter. These disks are encased in a rigid envelope. Despite their small size these have a larger storage capacity than the older floppy disks.

Figure 4.6: Floppy disk

A **floppy disk** is a magnetic storage medium that consists of a circular piece of thin, flexible (hence the name floppy) magnetic media encased in a square or rectangular plastic wallet. Floppy disks are read and written by a floppy disk drive or FDD.

Unlike most **hard disks**, the floppy disks are portable. Floppy disks are slower to access than hard disks and have less storage capacity, but they are much less expensive. Floppies come in three basic sizes i.e. 8-inch, 5¼-inch, 3½-inch, but the last one is the most commonly used.

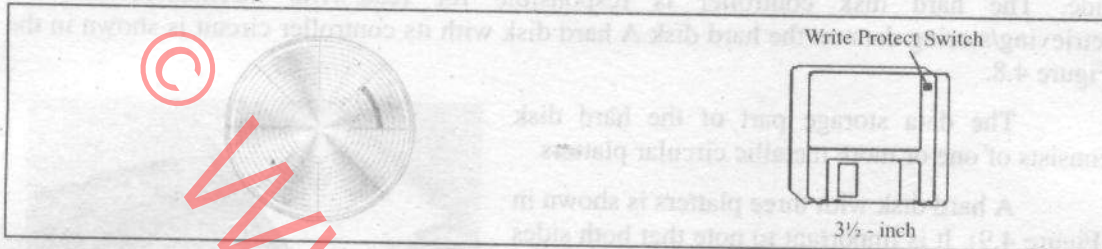



Figure 4.7: Floppy disk – internal view

The following is the series of actions taking place when data is written on the disk

- The computer program passes an instruction to the computer hardware to write a data file on a floppy disk.
- The computer hardware and the floppy-disk-drive controller start the motor in the diskette drive to spin the floppy disk.
- A second motor, called a stepper motor, rotates a worm-gear shaft in minute increments that match the spacing between tracks.
- The read/write heads stop at the track. The read head checks the prewritten address on the formatted diskette to be sure it is using the correct side of the diskette and is at the proper track.
- Then the data is written to the required address.
- The diskette stops spinning. The floppy disk drive waits for the next command.
- On a typical floppy disk drive, the small indicator light stays on during all of the above operations.

4.5.2 Hard Disk

Most digital computers use atleast one hard-disk drive. Some large scale computers normally contain hundreds of hard disks. Hard disks are used to permanently store digital data so you can say that hard disks give computers the ability to remember things when the power goes out. In this section we shall learn the function of a hard disk and also analyze the working of a hard disk.

CAPACITY AND PERFORMANCE

Nowadays a typical desktop computer has a hard disk with a capacity of more than 80 gigabytes. Data is stored onto the disk in the form of files. A file is simply a named collection of bytes. The bytes might be the ASCII (American Standard Code for Information Interchange) codes for the characters of a text file, or they could be the instructions of a software application for the computer, or they could be the stored information, or they could be the pixel colors for an image. There are two ways to measure the performance of a hard disk.

Data rate - The data rate is the number of bytes per second that the drive can deliver to the CPU. Rates between 5 and 40 megabytes per second are common.

Seek time - The time used to move the head to the appropriate track after reading the address is called the **seek time**

A typical hard disk consists of a sealed metallic box with controller circuit on one side. The hard disk controller is responsible for read/write mechanism and for retrieving/storing data on the hard disk A hard disk with its controller circuit is shown in the Figure 4.8.

The data storage part of the hard disk consists of one or more metallic circular platters

A hard disk with three platters is shown in (Figure 4.9). It is important to note that both sides of the platter have their own read/write head. The hard disk controller uses these heads to store and retrieve data from the disk. By arranging data on multiple platters the performance of the hard disk increases.

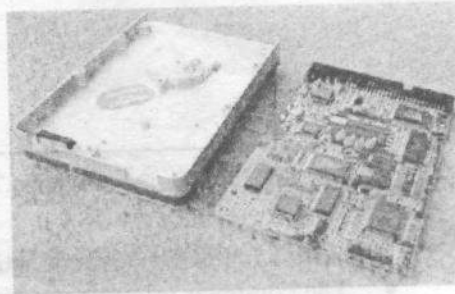



Figure 4.8: Hard disk – internal view

DATA ORGANIZATION

Data is stored on the surface of a platter in **sectors** and **tracks**. Tracks are concentric circles. The tracks are further divided into sectors. As shown in Figure 4.10 the yellow circle is a track and the blue part represent one sector. Typically a track is divided into 8 sectors. A sector usually contains a fixed number of bytes of data i.e. 512 bytes. When data is to be retrieved from the hard disk the operating system of the computer usually reads the whole track into the memory even if only one byte is needed. This usually increases the performance of the computer system.

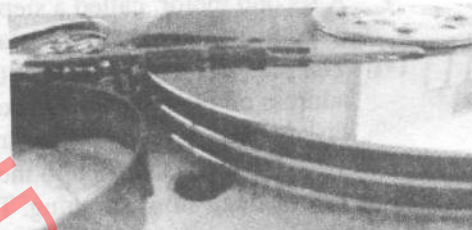



Figure 4.9 Hard disk

As we learned earlier that a hard disk can have more than one platter and each platter have two surfaces. The Tracks on a surface are numbered from 0,1,2 …n . All the tracks on the disk with same track number make up a cylinder.

It is important to note that the position of tracks and sectors are not fixed but these positions are marked by a process called format. Format is of two different types:

LOW LEVEL FORMATTING

During the process of low-level formatting, a drive marks the tracks and sectors on the disk. Usually this is done by the manufacturer of the disk. In this process the starting and ending points of each sector are written onto the platter. This process prepares the drive to hold data.

Figure 4.10 Tracks and sectors

HIGH LEVEL FORMATTING

During high-level formatting, the information about file-storage is written onto the disk called file-allocation table. This process also prepares the drive to hold data.

HOW DATA IS STORED ON/RETRIEVED FROM THE HARD DISK

As shown above the data is organized into tracks and sectors. Each track has a unique number. First track always has the number 0 0 0 called track zero. Similarly sectors on a track are numbered. When some software or operating system of the computer wants to read some data on some part of the disk it specifies the address of the location and provides the data. By using the provided address, the disk controller moves the read/write heads to the required track. It also uses the motor in the disk to rotate the disk platters. Because of this mechanical component this process is very slow as compared to the speed of the processor. When the head reaches the required track the read/write head has to wait for some time so that the required sector comes under it due to the rotation of the platters. This delay is called the **rotational delay**. When the appropriate sector comes under the read/write head it reads the data from the disk and send this data to the processor. The time consumed in this process is called the **transfer delay**. These three delays are used to calculate the access time of data.

Access Time = Seek Time + Rotational Delay + Transfer Delay

Obviously the seek time and rotational delay involve mechanical parts and are very large. Because of the delays the hard disk is very slow as compared to the CPU.

4.5.3 Compact Disks

One of the most prominent optical storage systems is the Compact Disk (CD), which is compatible with those in the music industry except that computer CD players spin the CD faster than the original CD's used in the music industry to obtain higher data transfer rates.

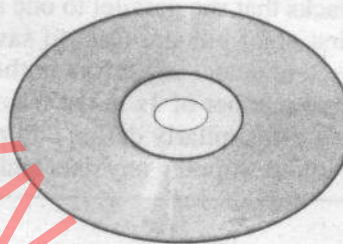



Figure 4.11

These disks are approximately 5 inches in diameter and consist of reflective material covered with a clear protective coating. Information is recorded on them by creating variations in their reflective surfaces. The information can then be retrieved by detecting these variations with a laser beam. Information on a CD is stored on one continuous track that spirals around the CD like a groove in an old-fashioned record. This is different from the magnetic disks where data is stored in concentric tracks.

The CD is commonly used to store data. CD is usually called CD-ROM (compact disk read-only memory). it can store more than 700 MB of data and are very useful for storing audio and video data. Following is a list of areas where the CD-ROM is used successfully for different purposes.

- Incorporate video on your CD-ROM to make an effective sales tool.
- Distributing different software products e.g., most operating systems are distributed on CD-ROM.
- Distributing audio and video data.
- Keeping the backup of large volumes of data and document archives
- Storing large volumes of data for uses in online application.

4.5.4 Tape Storage

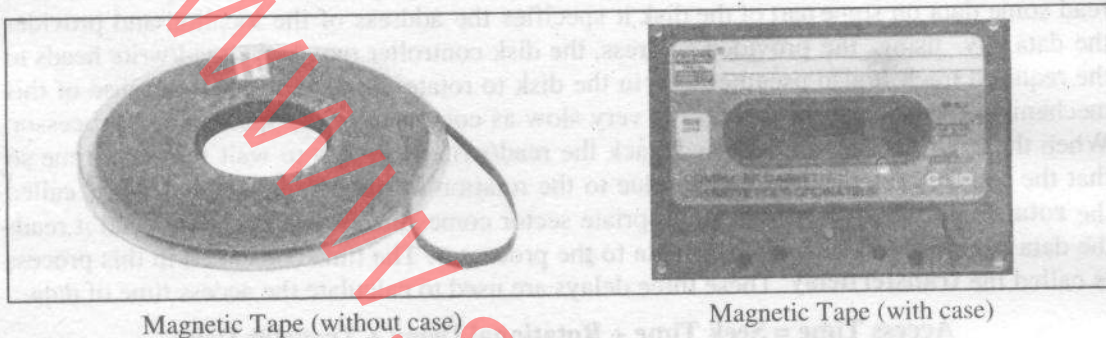
This is an old form of mass storage device that uses magnetic tape. In a magnetic tape information is recorded on the magnetic coating of a plastic tape. To access the data, this tape is mounted in a device called a tape drive that can read, write, and rewind the tape . Tape drives have different sizes ranging from very small cartridge units that use tape similar in appearance to that in stereo systems to large reel-to-reel units. The capacities of these devices vary a lot and some tapes can hold several gigabytes of data.

Magnetic Tape (without case) Magnetic Tape (with case)

Figure 4.12:

HOW DATA IS ORGANIZED ON A MAGNETIC TAPE

Modern streaming tape systems divide a tape into segments, each of which is magnetically marked by a gap when we format the disk. Each of these segments contains several tracks that run parallel to one another lengthwise on the tape this is shown in Figure 4.13. The first eight bits are used to save data and the last bit is used to store parity bit. This bit is used to detect any errors in the data stored on the tape. If this bit is set to 1 or zero so that the total number of 1s in the frame is even. This method of detecting error is called even parity. We can similarly define odd parity. The inter block gaps are needed so that the tape can stop without skipping any data and can be accelerated before reading data.

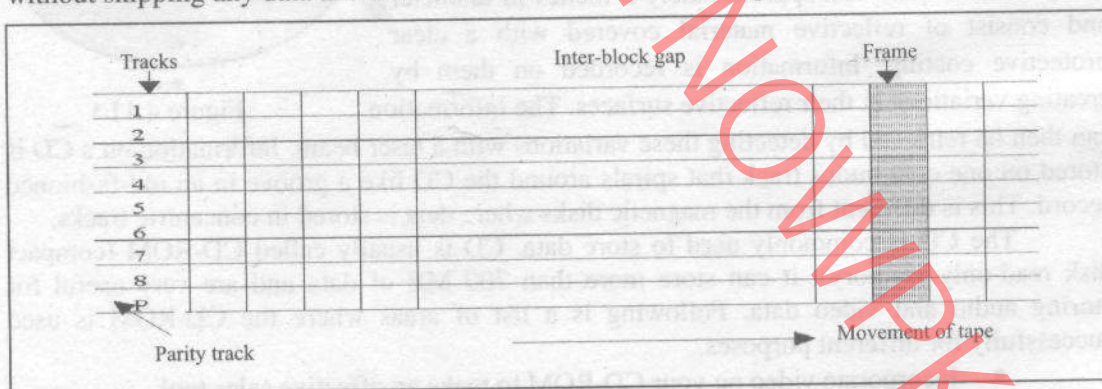



Figure 4.13: Data organization on magnetic tape

Data stored on a magnetic tape can be accessed only sequentially. This is the main disadvantage of streaming tape systems because moving between different positions on a tape can be very time-consuming. Thus tape systems have much longer data access times than disk systems in which different sectors can be accessed by short movements of the read / write

head. The tape systems are not popular for on-line data storage. On the other hand these tape devices are very cheep as compared to the magnetic disks. A large volumes of data can be stored on the tapes for backup purposes so these are used mainly in off-line, backup storage applications.

Exercise

1. Describe in detail the purpose and working of the main memory.
2. Describe in detail the purpose and working of the following Secondary storage devices.
   a. Floppy disk
   b. Hard Disk
3. Describe in detail the purpose and working of the following backing storage devices.
   a. Compact Disks
   b. Magnetic Tape
4. Explain, using a labeled diagram, the concepts of track and sector when describing magnetic disk storage.
5. Explain the purpose of the following and draw a diagram showing their relationship.
   a. Cache memory
   b. Hard disk
   c. Magnetic Tape
6. Explain why secondary memory is needed in a computer system?
7. Explain the purpose of following:
   i. High level formatting
   ii. Low level formatting
   iii. RAM and ROM
8. A 9th class student has a home computer system. What storage devices, the student will use on the home computer system. Explain why these devices are needed?
9. Fill in the blanks :
   (i) ______________ is a direct access storage device.
   (ii) ______________ is a serial access storage device.
   (iii) Access time = ______________ time + ______________ time .
   (iv) RAM stands for ______________
   (v) 1 MB is equal to ______________ bytes.
   (vi) The contents of ______________ must be refreshed periodically.
   (vii) The time required to move the head of the hard disk to appropriate track is called ______________.
   (viii) The larger the size of the RAM, the __________ the efficiency of the computer.
   (ix) EPROM stands for ______________ .
   (x) MSB stands for ______________ .

head. The tape systems are not popular for on-line data storage. On the other hand these tape devices are very cheep as compared to the magnetic disks. A large volumes of data can be stored on the tapes for backup purposes so these are used mainly in off-line, backup storage applications.

Exercise

1. Describe in detail the purpose and working of the main memory.
2. Describe in detail the purpose and working of the following Secondary storage devices.
   - a. Floppy disk
   - b. Hard Disk
3. Describe in detail the purpose and working of the following backing storage devices.
   - a. Compact Disks
   - b. Magnetic Tape
4. Explain, using a labeled diagram, the concepts of track and sector when describing magnetic disk storage.
5. Explain the purpose of the following and draw a diagram showing their relationship.
   - a. Cache memory
   - b. Hard disk
   - c. Magnetic Tape
6. Explain why secondary memory is needed in a computer system?
7. Explain the purpose of following:
   - i. High level formatting
   - ii. Low level formatting
   - iii. RAM and ROM
8. A 9th class student has a home computer system. What storage devices, the student will use on the home computer system. Explain why these devices are needed?
9. Fill in the blanks :
   - (i) ________________ is a direct access storage device.
   - (ii) ________________ is a serial access storage device.
   - (iii) Access time = ______________ time + __________________ time .
   - (iv) RAM stands for ____ .
   - (v) 1 MB is equal to ________________bytes.
   - (vi) The contents of _______________ must be refreshed periodically.
   - (vii) The time required to move the head of the hard disk to appropriate track is called ______________.
   - (viii) The larger the size of the RAM, the _________ the efficiency of the computer.
   - (ix) EPROM stands for ________________ .
   - (x) MSB stands for ________________ .

10. Match the following :

Hard disk	Serial access
RAM	Secondary storage device
Tape storage	Optical storage
CD	Primary storage

11. Choose the correct answer

a. Tape storage is
(i) Slower than the hard disk. (ii) Faster than hard disk.
(iii) Direct access device. (iv) All above.

b. 1 KB is equal to
(i) 1000 bytes (ii) 2^{10} bytes (iii) 2^{20} bytes (iv) 2^{30} bytes

c. Cache memory is
(i) Faster than the main memory. (ii) Slower than the main memory.
(iii) Smaller than the main memory. (iv) Only(i) and (iii). (v) None of the above.

d. Impact printers
(i) Touch the surface of the paper during printing process.
(ii) Don't touch the surface of the paper during printing process.
(iii) Faster than non impact printers. (iv) All of the above.

e. Static RAM
(i) contents are required to be refreshed periodically.
(ii) contents are not required to be refreshed periodically.
(iii) is faster than DRAM. (iv) only(i) and (ii). (v) only(ii) and (iii)

12. Mark the following as True/False

(i) Tape storage is a direct access storage device (ii) ROM is volatile
(iii) SRAM is faster than DRAM. (iv) 1 MB = 2^{20} bytes
(v) every program must be loaded into RAM before execution
(vi) Zero sector of floppy disk consists of seven tracks
(vii) System Bus is the pathway among different components of computer
(viii) Secondary storage is cheaper than primary storage
(ix) CD is a magnetic storage device
(x) In floating point number format, the exponent is represented in signed magnitude form.

Answers

Q.9
(i) Hard Disk (ii) Magnetic Tape (iii) Seek, Latency (iv) Random Access Memory
(v) 2^{20} (vi) DRAM (vii) Seek Time (viii) Higher
(ix) Erasable Programmable Read Only Memory (x) Most Significant Bit

Q.11
a. i b. ii c. iv d. i e. v

Q.12
(i) F (ii) F (iii) T (iv) T (v) T
(vi) F (vii) T (viii) T (ix) F (x) F

Chapter 5

Number Systems

5.1 Data and Information

Data is defined as the collection of facts and figures, whereas the data after processing is called **information.**

Suppose that fifteen students of a class appear in an examination. You are assigned the task of calculating the pass percentage of the overall class and the percentage and grade of each of the students (total masks are 550).

How would you find out the required information? The first step is the data collection; you note down the marks of all students of the class, suppose these are 354, 285, 421, 360, 298, 159, 163, 148, 270, 467, 305, 221, 341, 255, and 311. You are also provided the roll numbers of the students. At this stage the above list of numbers represent **data** you have collected to find the required information. This data is in its raw form; it does not provide the required information. You will have to process it by keeping in mind the required information. Processing may involve certain steps such as applying certain calculations, sorting and formatting etc.

By applying certain calculations we get the following table:

Roll No.	Marks	%age	Grade	Overall pass %age of the class
1	354	64.36	B	
2	285	51.82	C	
3	421	76.55	A	
4	360	65.45	B	
5	298	54.18	C	
6	159	28.91	F	
7	163	29.64	F	
8	148	26.91	F	80%
9	270	49.09	D	
10	467	84.91	A+	
11	305	55.45	C	
12	221	40.18	D	
13	341	62.00	B	
14	255	46.36	D	
15	311	56.55	C	

The above table shows the information extracted from the collected data. It shows clearly, when the data undergoes certain processing we get information. The processing can

be done manually or by using a computer. Computer processes data in terms of binary numbers i.e.,0 or 1.

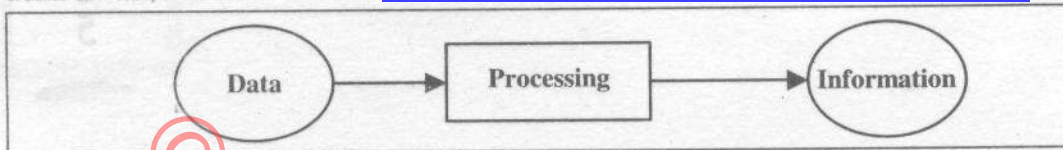



Figure 5.1

Both data and information may be represented in many forms e.g. text, sounds, pictures, graphs, etc. It is useful to observe that a computer is just a data processing machine. It inputs data, perform processing and then output the processed data i.e. information. It is important to know about the different types of data and how these data are represented within a computer.

All computer programs use one or more of the following types of data.

Numeric data, Alphabet data, Alphanumeric data.

5.1.1 Numeric data

Numeric data is used to represent different quantities on which arithmetic is to be performed e.g. marks of different students, sales records of goods at a shop etc. Mostly this data is represented as integers or real numbers e.g. 40, 323, -76.07 etc. there are two types of numeric data:

- Integer
- Real number

5.1.2 Alphabetic data

Alphabet data only consists of a fixed set of alphabetic characters e.g. data consisting of English alphabets A, B, C... Z as well as a, b, c... z. We can use these English alphabets to represent names of students in a class. This data is represented as a sequence of characters and no arithmetic operations can be carried out on it.

5.1.3 Alphanumeric data

Alphanumeric data contains alphabets, numbers and other special characters i.e. $, #, % etc. Example of such data can be telephone numbers and addresses such as, (092) 051-2345682 and House # 967, street 9, ABC colony, Rawalpindi etc.

5.2 Number Systems

A number system defines a set of values used to represent different quantities. For example; we can represent the number of students in our class or number of viewers watching a certain TV program etc. It is interesting to note that in a digital computer all kinds of information and data (e.g. audio, Graphics, Video, Text and Numeric) are represented as binary numbers. In general, decimal system is used while the computers use binary number system. Octal and Hexadecimal numbers systems are also used commonly in computer system that's why, we need a conversion from one number system to another.

5.2.1 Decimal (Base 10) Number System

We are familiar with ten digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and also know that any value can be represented by using these ten digits only. For example, the value four hundred and fifty three can be written as:

> Do you remember $10^0 = 1$?
> What do you know about $N^0 = 1$ where N is not 0?

$$453 = 4 \times 10^2 + 5 \times 10^1 + 3 \times 10^0.$$

We can also write numbers with fractional parts in this system by using only ten digits and the decimal point (.) e.g. 139.78 can be written as,

$$139.78 = 1\times10^2 + 3\times10^1 + 9 \times 10^0 + 7 \times 10^{-1} + 8 \times 10^{-2}$$

It is a positional number system that means the position of a digit within a number is very important so that 39 and 93 represent two different values.

> Can you name a non-positional number system?

In this system of numbers each number consists of digits located at different positions with position 0 is the first digit towards the left side of the decimal point, position 1 is the second digit towards the left side of the decimal point and so on. Similarly the first digit towards the right side of the decimal point is at position –1, second has position -2 and so on. Each position has a weight assigned to it e.g. position 1 has weight 10^1 position 2 has weight 10^2 and so on. This is shown in the table below:

> Note that positions have weights in powers of 10 because we have 10 digits in the number system

Table demonstrating the position and weights of digits for the number 57231.21

Position	4	3	2	1	0	-1	-2
Face Value	5	7	2	3	1	2	1
Weights	10^4	10^3	10^2	10^1	10^0	10^{-1}	10^{-2}

or, $57231.21 = 5\times 10^4 + 7 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 1 \times 10^0 + 2 \times 10^{-1} + 1 \times 10^{-2}$

It is clear from the above discussion that the value of the number is determined by multiplying the digits with the weight of their position and adding the results. This method is called the expansion method. The digit at the extreme right side of a number is called the Least Significant Digit (LSD) because it has least weight and the digit at the extreme left side of a number is called the Most Significant Digit (MSD) because it has maximum weight. For example; in the number 724, 7 is most significant and 4 is least significant.

5.2.2 Binary Number System

This number system uses only two digits 0 and 1 to represent any quantity. These digits are called Binary digits or BIT. Like the decimal number system this is also a positional number system and each position has a weight that is a power of 2. For example

> Why do we use powers of 2 in the expansion of a binary number?

$$01001_{(2)} = 0\times2^4 + 1\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0 = 9_{(10)}$$

The position 0 has weight $2^0 = 1$ and position 1 has weight 2^1. Similarly, we can also represent fractional binary numbers as shown below:

$$101.101_{(2)} = 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} = 5.625_{(10)}$$

Table demonstrating the position and weights of digits for the number $101.101_{(2)}$

Position	2	1	0		-1	-2	-3
Face Value	1	0	1	.	1	0	1
Weights	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}

5.2.3 Hexadecimal Number System

You may have observed that binary numbers are not very easy to work with because even very small numbers $277_{(10)}$ need at least 9 bits i.e., $277_{(10)} = 0100010101_{(2)}$ and also converting binary to decimal and decimal to binary needs much calculation. Because of these difficulties computer scientists use another number system frequently. This number system is base 16 or hexadecimal number system .This number system uses sixteen different digits. The digits are:

Remember that
A = Ten
B = Eleven
C = Twelve
D = Thirteen
E = Fourteen
F = Fifteen.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

$758_{(16)}$ is a hexadecimal number and is different from $758_{(10)}$ which is seven hundred and fifty seven. You may read $758_{(16)}$ as **Seven Five Eight Base Sixteen** and not as seven hundred and fifty eight. A fractional hexadecimal number e.g.,$758.D1_{(16)}$ can be represented as shown below:

$$758.D1_{(16)} = 7\times16^2 + 5\times16^1 + 8\times16^0 + D\times16^{-1} + 1\times16^{-2} = 1880.8164_{(10)}$$

(up to four decimal places)

Table demonstrating the position and weights of digits for the number $758.D1_{(16)}$

Position	2	1	0		-1	-2
Face Value	7	5	8	.	D	1
Weights	16^2	16^1	16^0		16^{-1}	16^{-2}

5.2.4 Octal Number System

This is another number system that can also be used in computers. This is the base-8 number system or octal number system. This number system uses only eight different digits to represent numbers. The digits are 0, 1, 2, 3, 4, 5, 6, 7. In this number system $751_{(8)}$ is a valid number and is different from seven hundred and fifty one. Clearly **821** can not be a number in this system as 8 is not a valid digit in this number system. An octal number e.g.,$630.4_{(8)}$ can be represented as shown below:

$$630.4_{(8)} = 6\times8^2 + 3\times8^1 + 0\times8^0 + 4\times8^{-1} = 408.5_{(10)}$$

Table demonstrating the position and weights of digits for the number $630.4_{(8)}$

Position	2	1	0		-1
Face Value	6	3	0	.	4
Weights	8^2	8^1	8^0		8^{-1}

5.3 Number System Conversion

We use decimal system and computer use binary system. Also the octal and hexadecimal number systems are frequently used in computerized data processing system. An

interesting problem is the conversion of data from one number system to another number system.

5.3.1 Conversion of Decimal Number into Binary

To convert a decimal number into Binary, we can use the repeated division procedure as shown in following example.

Example: Convert 27 into binary

Solution:

	Number	Remainder
2	27	
2	13	1
2	6	1
2	3	0
2	1	1
	0	1

We should stop the division process when 0 is the answer of some division and collect the remainders in reverse order as is indicated by the arrow:

So $27_{(10)} = 011011_{(2)}$

5.3.2 Converting Fractional Decimal Numbers to Binary

Following example demonstrates the process of converting a decimal fraction in binary:

Example: Convert 0.56 into binary. Give answer up to 6 decimals

Solution:

			Result	Fractional part	Integral part
2	×	0.56	1.12	12	1
2	×	0.12	0.24	24	0
2	×	0.24	0.48	48	0
2	×	0.48	0.96	96	0
2	×	0.96	1.92	92	1
2	×	0.92	1.84	84	1
2	×	0.84	1.68	68	1
2	×	0.68	1.36	36	1

$0.56_{(10)} = 0.100011_{(2)}$

The bits to the left of point are obtained by using the integer parts of the answers in the order shown by the arrow. Note that we continue this process of multiplication until we get 0 or a repeated value as answer or get required number of bits.

5.3.3 Converting Real Numbers into Binary

Next example shows the process of converting a real number (i.e., a number that has both an integer part and a fractional part) into binary.

Example: Convert $56.25_{(10)}$ into binary

Solution: To convert this real number we independently convert the integral part (i.e, 56) and the fractional part (i.e.,0.25) into binary by using the processes given above

	Number	Remainder
2	56	
2	28	0
2	14	0
2	7	0
2	3	1
2	1	1
	0	1

$56 = 0111000_{(2)}$

			Result	Fractional part	Integral part
2	×	0.25	0.5	5	0
2	×	0.5	1.0	0	1

$0.25 = .01_{(2)}$

so, $\mathbf{56.25 = 0111000.01_{(2)}}$

Note: The above result is obtained by connecting the two results.

5.3.4 Converting Binary Numbers into Decimal

As is shown in the examples below we can use the expansion method to convert a binary number into decimal system.

Example 1: Convert $011011_{(2)}$ into decimal

Solution: $011011_{(2)} = 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 27_{(10)}$

Example 2: Convert $1110.11_{(2)}$ into decimal

Solution: $1110.11_{(2)} = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$

$= 8 + 4 + 2 + 0 + 1/2 + 1/4 = 14.75$

5.3.5 Conversion of Decimal into Hexadecimal

The process of converting a decimal number into hexadecimal is shown in the examples below.

Example 1: Convert $185_{(10)}$ into hexadecimal

Solution:

	Number	Remainder
16	185	
16	11	9
	0	B

So $\mathbf{185_{(10)} = 0B9_{(16)}}$

Example 2: Convert $0.3_{(10)}$ into hexadecimal.

Solution:

			Result	Fractional part	Integral part
16	×	0.3	4.8	8	4
16	×	0.8	12.8	8	12=C
16	×	0.8	12.8	8	12=C

So $0.3_{(10)} = 0.4C_{(16)}$ (because C is the repeating value therefore, as a Convention, we shall take it once only)

Example 3: Convert $185.3_{(10)}$ into hexadecimal

Solution: As given in the above two examples

$185_{(10)} = B9_{(16)}$ and $0.3_{(10)} = 0.4C_{(16)}$

So $185.3 = 0B9.4C_{(16)}$

As you can see the conversion of a decimal to hexadecimal is similar to the conversion of decimal into Binary. Actually this repeated division or repeated multiplication method can be used for converting a decimal number into any other base.

5.3.6 Conversion of Hexadecimal into Decimal

We can also convert a hexadecimal number into decimal by using the expansion method shown in the following examples :

Example 1: Convert $0B9_{(16)}$ into decimal

Solution: $0B9_{(16)} = 0 \times 16^2 + B \times 16^1 + 9 \times 16^0 = 0 \times 16^2 + 11 \times 16^1 + 9 \times 16^0 = 185_{(10)}$

Example 2: Convert $0B9.4C_{(16)}$ into decimal

Solution:

$$\begin{aligned} 0B9.4C_{(16)} &= 0 \times 16^2 + B \times 16^1 + 9 \times 16^0 + 4 \times 16^{-1} + C \times 16^{-2} \\ &= 0 \times 16^2 + 11 \times 16^1 + 9 \times 16^0 + 4 \times 16^{-1} + 12 \times 16^{-2} \\ &= 0 + 176 + 9 + 4/16 + 12/256 \\ &= 0 + 176 + 9 + 1/4 + 3/64 = 185.296875_{(10)} \end{aligned}$$

5.3.7 Conversion of Hexadecimal into Binary

As was stated before that the conversion between binary and decimal numbers is a tedious process. On the other hand the conversion of a hexadecimal to Binary is very easy and is shown in the example below.

Example 1: Convert $10A8_{(16)}$ into Binary

Solution:

Step 1: Convert each digit into binary separately and write it in 4 bits.

$1 = 0001_{(2)}$

$0 = 0000_{(2)}$

$A = 1010_{(2)}$

$8 = 1000_{(2)}$

Step 2: Replace the digits of the hexadecimal number with the four bits obtained in step 1.

$$10A8_{(16)} = 0001\ 0000\ 1010\ 1000_{(2)}$$

Example 2: Convert $A1.03_{(16)}$ into Binary

Solution:

Step 1: Convert each digit independently into binary and write it in 4 bits.

A = $1010_{(2)}$
1 = $0001_{(2)}$
0 = $0000_{(2)}$
3 = $0011_{(2)}$

Step 2: Replace the digits of the hexadecimal number with the four bits obtained in step 1.

$$10A8_{(16)} = 1010\ 0001\ .0000\ 0011_{(2)}$$

5.3.8 Conversion of Binary into Hexadecimal

The conversion of binary to hexadecimal is the reverse of conversion of hexadecimal into binary and is shown in the example below.

Example 1: Convert $10010011_{(2)}$ into hexadecimal.
Solution:

Step 1: First divide your number into groups of 4 bits starting from the right side
$10010011_{(2)}$ is divided into the following two groups 1001 and 0011

Step 2: Convert each group into hexadecimal.
$1001=9_{(16)}$ and $0011=3_{(16)}$

Step 3: Replace each group by its hexadecimal equivalent.

$$1001\ 0011_{(2)} = 93_{(16)}$$

Example 2: Convert $101100.1_{(2)}$ into hexadecimal.

Solution:

Step 1: First divide your number into groups of 4 bits starting from the decimal point position so $101100.1_{(2)}$ is divided into the following three groups 1100 and 0010 on the left of decimal point and 1000 on the right of decimal point

Step 2: Convert each group into hexadecimal.
$1100= 12 = C_{(16)}$, $0010 = 2_{(16)}$ and $1000=8_{(16)}$

Step 3: Replace each group by its hexadecimal equivalent.

$$10\ 11\ 00.1_{(2)} = 2C.8_{(16)}$$

Explain!
Why do we make 4 bit groups?

Note that if the last group on the left of decimal has less than 4 bits then extra 0 bits are added at the left end of the number. Similarly if the last group on the right of decimal has less than 4 bits then extra 0 bits are added at the right end.

The following table can be used to convert any hexadecimal number into binary

Table for Hexadecimal to Binary Conversion

Hexadecimal Number	Binary Equivalent	Hexadecimal Number	Binary Equivalent
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

5.3.9 Conversion of Decimal into Octal

The repeated division method will be used to convert the decimal number into octal and is shown in the following example.

Example 1: Convert $185_{(10)}$ into Octal

Solution:

		R
8	185	
8	23	1
8	2	7
8	0	2

So $185_{(10)} = 0271_{(8)}$

We can also convert a fractional octal number into decimal by using the expansion method shown in the next example

Example 2: Convert $0.3_{(10)}$ into octal and give answer up to 5 decimal places.

Solution:

8×0.3	=	2.4	0.4	2
8×0.4	=	3.2	0.2	3
8×0.2	=	1.6	0.6	1
8×0.6	=	4.8	0.8	4
8×0.8	=	6.4	0.4	6

So $0.3_{(10)} = 0.23146_{(8)}$

Example 3: Convert $185.3_{(10)}$ into octal . Give answer up to 5 decimal places.

Solution: As given in the two examples above

$185_{(10)} = 0271_{(8)}$ and $0.3_{(10)} = 0.23146_{(16)}$

So $185.3 = 0271.23146_{(8)}$

5.3.10 Conversion of Octal into Decimal

Example 1: Convert $0271_{(8)}$ into decimal

Solution: $0271_{(8)} = 0 \times 8^3 + 2 \times 8^2 + 7 \times 8^1 + 1 \times 8^0 = 185_{(10)}$

Example 2: Convert 271.231$_{(8)}$ into decimal

Solution: $0271.231_{(8)} = 0 \times 8^3 + 2 \times 8^2 + 7 \times 8^1 + 1 \times 8^0 + 2 \times 8^{-1} + 3 \times 8^{-2} + 1 \times 8^{-3}$

$= 0 + 128 + 56 + 1 \quad + 2/8 + 3/64 + 1/512$

$= 185.2988_{(10)}$

5.3.11 Conversion of Octal Number into Binary

The conversion of an octal number to binary is similar to the conversion of hexadecimal into binary and shown in the following example.

Example 1: Convert 107$_{(8)}$ into Binary

Step 1: Convert each digit independently into binary and write it in 3 bits.

$1 = 001_{(2)}$
$0 = 000_{(2)}$
$7 = 111_{(2)}$

Step 2: Replace the digits of the hexadecimal number with the three bits obtained in step 1.

$107_{(8)} = 001\ 000\ 111_{(2)}$

Example 2: Convert 107.52$_{(8)}$ into Binary

Solution:

Step 1: Convert each digit independently into binary and write it in 3 bits.

$1 = 001_{(2)}$ $\quad 0 = 000_{(2)}$ $\quad 7 = 111_{(2)}$
$5 = 101_{(2)}$ $\quad 2 = 010_{(2)}$

Step 2: Replace the digits of the octal number with the three bits obtained in step 1

$107.52_{(8)} = 001\ 000\ 111.\ 101\ 010_{(2)}$

5.3.12 Conversion of Binary into Octal

The conversion of binary number to octal can be done directly and is shown in the example below

Example 1: Convert 10010011$_{(2)}$ into octal

Solution:

Step 1: First divide your number into groups of 3 bits starting from the right side

So $10010011_{(2)}$ is divided into the following three groups

010, 010 and 011

Step 2: Convert each group into octal.

$010 = 2_{(8)};$ $\quad 010 = 2_{(8)};$ $\quad 011 = 3_{(8)}$

Step 3: Replace each group by its octal equivalent.

$010\ 010\ 011_{(2)} = 223_{(8)}$

Example 2: Convert 11010.11$_{(2)}$ into octal

Solution:

Step 1: First divide your number into groups of 3 bits starting from the right side

So $11010.11_{(2)}$ is divided into the following three groups
010, 011 on the left of decimal and 110 on the right of decimal

Can you guess the reason of making groups of 3 Bits?

Step 2: Convert each group into octal.

$010 = 2_{(8)}$, $011 = 3_{(8)}$, $110 = 6_{(8)}$

Step 3: Replace each group by its octal equivalent.

$100100.110_{(2)} = 32.6_{(8)}$

Note: If the last group has less than 3 bits then add extra 0 bits at the left or right end depending if it is on the left or right side of the decimal point, respectively.

Table for Octal to Binary Conversion

Octal Number	Binary Equivalent	Octal Number	Binary Equivalent
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

5.4 Representation of Numbers using 1's and 2's Complements

Representing Signed Numbers

Now that you know how to represent positive numbers in different number systems for example. Base 2, Base 8, Base 10 and Base 16. It is time to look at another interesting question.

How do we represent both positive and negative numbers in Binary number system?

There are many methods of representing signed numbers in binary e.g. **sign–magnitude** method, **1's complement** method, **2's complement** method and **Access notation**. In this section we will study 1's complement method and 2's complement methods of representing signed numbers. These two methods are very useful when we want to perform binary arithmetic.

5.4.1 1's Complement Method

To understand this method of representing signed numbers we first see the meaning of the 1's complement of a binary number.

1's complement of an 8-bit binary number is obtained by subtracting the number from $11111111_{(2)}$ as shown in the example below.

Example 1: **Take 1's complement of the binary number 10011001**

Solution:

```
                     11111111
                    -10011001
1's Complement       01100110
```

Observe that 1's complement of a binary number can be directly obtained by changing all 0's to 1's and all 1's to 0's

Example 2: Take 1's complement of the binary number 01100110 directly.

Solution:

Original number	01100110
1's Complement	10011001

Representation of Negative numbers using 1's complement:

To represent the negative numbers in 1's complement form, we perform the following steps:

- First determine the number of bits to represent the number .
- Convert the modules of the given number in binary.
- Place a 0 in MSB and binary conversion of the number in remaining bits MSB (Most Significant Bit).
- Take 1's complement of the result.

Example 3: Represent -54$_{(10)}$ in 1's complement form using 8 bits .

Solution:

Number of bits = 8

54$_{(10)}$ = 0110110$_{(2)}$=00110110$_{(2)}$(in 8 bits)

So-54 in 1's complement form = 11001001

So 1's complement representation of negative integers is same as the usual binary number representation and will always have a 0 in MSB.

5.4.2 2's Complement Method

We know that most computers use 16 bits to represent integers. When numbers are represented in fixed number of bits the 2's complement method is a very useful way of representing signed number. Most computers represent integers using this method. Many digital calculators also used this method for representing integers.

2's complement of a binary number can be obtained by first taking 1's complement and then adding 1 in the result. This process is shown in the example below:

Example 1: Take 2's complement of the binary number 01100110$_{(2)}$.

Solution:

Step 1: Taking 1's complement of the given number results in 10011001.

Step 2: Adding 1 in the result give us

```
  10011001
        +1
  --------
  10011010
```

2's complement of 01100110 = 10011010

We can obtain 2's complement of a binary number directly without taking 1's complement. To take 2's complement directly copy the number without any change up to the first (least significant) 1 in the number and change reaming 0s to 1s and 1s to 0s. This process is shown in the example below.

Example 2: Take 2's complement of the binary number 01100110$_{(2)}$ directly.

Given number: 01100110

2's Complement: 10011010

Representation of Negative numbers using 2's complement

To represent the negative numbers in 2's complement form we perform the following steps:

- First determine the number of bits to represent the number
- Convert the modules of the given number in binary.

- Place a 0 in MSB and binary of the number in remaining bits.
- Take 2's complement of the result.

This is shown in the example given below.

Example 3: Represent -54$_{(10)}$ in 2's complement form using 8 bits.

Solution:

Number of bits = 8
Modulus of -54= 54 = 0110110
54 in 2's complement form = 00110110
-54 in 2's complement form = 11001010

Clearly 2's complement representation of a negative integer will alway have a 1 in the most significant bit. The smallest integer in 2's complement form using 8 bits is 10000000 = -128 = -2^7 and the largest integer in 2's complement form using **n** bits is $2^{(n-1)}$

5.5 Binary Arithmetic

In this section we are going to learn the basic arithmetic operations i.e. addition, subtraction, multiplication and division for binary integers. Following table shows the results of performing the addition operations on two **bits.** This table can also be used to add two multi-bit binary numbers.

5.5.1 Binary Addition

Following table shows the results of performing the addition operations on two **bits.** This table can also be used to add two multi-bit binary numbers.

Operation	Result
0 + 0	0
0 + 1	1
1 + 0	1
1 + 1	0 **with a carry of 1**

The following example shows the process of adding two binary numbers by using the results from the table above.

Example: Add 01011101$_{(2)}$ and 00110010$_{(2)}$.

Solution:

```
Carry     1 1 1
          0 1 0 1 1 1 0 1
   +      0 0 1 1 0 0 1 0
          _______________
          1 0 0 0 1 1 1 1
```

Note that 1+1 should be 2. But in binary system 1+1 is 0 and a carry of one is generated.

Challenge

Can you guess 1+1+1 =?

Clearly the process of adding two binary numbers is similar to that of adding two decimal numbers. But in this process we use the rules given in the table above.

5.5.2 Binary Subtraction

The process of subtracting two binary numbers is also similar to decimal numbers subtraction. Following table shows the rules of subtracting two bits.

Operation	Result
0 – 0	0
0 – 1	1 with a barrow from the next position
1 – 0	1
1 – 1	0

Now that you know the rules for subtracting two bits, let us learn the process of subtracting two binary numbers through the following example.

Example: Subtract $01011101_{(2)}$ from $10110010_{(2)}$.

> **Observation**
> By using 1's and 2's complement method, we can perform subtraction by using addition only.

Solution:

Barrow

Note that when you take a barrow from the higher position 0 becomes $10_{(2)}$ and $10_{(2)} - 1_{(2)} = 1$

As it is difficult and expensive to make a computer that uses this method for performing subtraction. Most day computers use 2's complement or 1's complement method for performing subtraction.

Subtraction Using 1's Complement

The following examples explain fully the process of subtraction using 8-bit 1's complement method .

Example 1: Calculate 38 – 29 using 8-bit 1's complement method .

Solution: Here 38 – 29 = 38 + (-29)

Step 1: Write magnitude of both the numbers in binary form using 8 bits.

38 = $00100110_{(2)}$ and 29 = $00011101_{(2)}$

Step 2: Represent negative numbers in 1's complement.

- 29 = 11100010

Step 3: Add the 1's complement representation.

```
                      00100110
                   +  11100010
                   -----------
End carry: 1          00001000
Add End carry: +             1
                   -----------
Answer:               00001001
```

So the answer is 0 0 0 0 1 0 0 1 and it is in 1's complement.

Step 4: Convert the 1's complement result into decimal

00001001 = 9

Example 2: Calculate 45 – 63 using 8-bit 1's complement method .

Solution:

We can write 45 – 63 = 45 + (-63)

Step 1: Write magnitude of both the numbers in 8 bits.

45 = $00101101_{(2)}$ and 63 = $00111111_{(2)}$

Step 2: Represent number in 1's complement

- 63 = 11000000

Step 3: Add the 1's complement representation

```
          00101101
        + 11000000
        ----------
End Carry: 0       11101101
Add End Carry: +          0
        ----------
Answer:            11101101
```

So the answer is 11101101 and it is in 1's complement

Step 4: Convert the 1's complement result into decimal

11101101 = - 00010010 = -18

Note: If the End carry in the sum is 0 then we do not need to perform step 4.

Example 3: Calculate (– 54 – 30) using 8 bits 1's complement method.

Solution: (First number is negative as well)

We can write -54 – 30 = (-54) + (-30)

Step 1: Write magnitude of both the numbers in 8 bits.

$54 = 00110110_{(2)}$ and $30 = 00011110_{(2)}$

Step 2: Represent both numbers in 1's complement form.

-54 = 11001001 and -30 = 11100001

Step 3: Add the 1's complement representation.

```
          11001001
        + 11100001
        ----------
End Carry:1        10101010
Add End Carry: +          1
        ----------
Answer:            10101011
```

So the answer is 10101011 and it is in 1's complement.

Step 4: Convert the 1's complement result into decimal.

As the MSB is 1 so it is a negative number so,

10101011 = -01010100 = -84

Note: 1's complement use addition operation twice for subtracting binary numbers. First for adding the numbers and then for adding the final carry.

Subtraction Using 2's Complement.

The process of subtracting two binary numbers using 2's complement is explained through the following examples. We are using the same questions given in the examples of 1's complement method to so that you can compare the two methods.

Example 1: Calculate 38 – 29 using 8-bit 2's complement method

Solution: Here 38 – 29 = 38 + (-29)

Step 1: write magnitude of both numbers in 8 bits.

38 = $00100110_{(2)}$ and 29 = $00011101_{(2)}$

Step 2: represent negative number in 2's complement.

- 29 = 11100011

Step 3: add the 2's complement representation and ignore the end carry.

```
              00100110
            + 11100011
End Carry 1   00001001
```

So the answer is 0 0 0 0 1 0 0 1 and it is in 2's complement.

Step 4: Convert the 2's complement result into decimal

00001001 = 9

In the following examples we are going to subtract a larger number from a smaller number.

Example 2: Calculate 45 – 63 using 8-bit 2's complement method

Solution:

We can write 45 – 63 = 45 + (-63)

Step 1: write magnitude of both the numbers in 8 bits.

45 = $00101101_{(2)}$ and 63 = $00111111_{(2)}$

Step 2: represent negative numbers in 2's complement.

- 63 = 11000001

Step 3: add the 2's complement representation and ignore the end carry.

```
              00101101
            + 11000001
End Carry 0   11101110
```

Note that when we calculate –97 – 85 using 8-bit 2's complement, the answer should be –182 but it comes out to be 74. Can you explain why?

So the answer is 1 1 1 0 1 1 1 0 and it is in 2's complement form.

Step 4: Convert the 2's complement result into decimal

11101110 = - 00010010 = -18

Example 3: Calculate (-54 – 30) using 8 bits 2's complement method.

Solution: We can write -54 – 30 = (-54) + (-30)

Step 1: write magnitude of both numbers in 8 bits.

54 = $00110110_{(2)}$ and 30 = $00011110_{(2)}$

Step 2: represent both numbers in 2's complement.

-54 = 11001010 and -30 = 11100010

Step 3: add the 2's complement representation and ignore the end carry.

```
              11001010
            + 11100010
End Carry 1   10101100
```

So the answer is 1 0 1 0 1 1 0 0 and it is in 2's complement.

Step 4: Convert the 2's complement result into decimal

10101100 = - 01010100 = -84

You may have noticed that by using 1's complement and 2's complement method we can perform subtraction by using the addition operation only. So, if some one the same makes a digital circuit for adding two binary numbers we can also subtract the binary numbers by using circuit and also if some digital computer can add two binary numbers, then it can subtract the numbers as well.

5.5.3 Binary Multiplication

In this section we will first learn how to multiply two unsigned binary integers by using the familiar multiplication process and then also see another very interesting way of multiplication.

Following table gives the basic multiplication rules for two bits
. confuses with the decimal point so, replace it with x.

Product	Answer
0.0	0
0.1	0
1.0	0
1.1	1

The following example describe the process of multiplying two 4-bit binary numbers.

Example: Calculate $0110_{(2)} \times 1011_{(2)}$

Solution:

```
Multiplicand:      0 1 1 0
Multiplier:        1 0 1 1
              ------------
                   0 1 1 0
                 0 1 1 0 ×
               0 0 0 0 × ×
             0 1 1 0 × × ×
             -------------
             1 0 0 0 0 1 0
```

Obviously this is the usual way of multiplying numbers.

5.5.4 Division of Binary Numbers

Binary division method is demonstrated in the following examples.

Example 1: Calculate $01001101_{(2)} \div 111_{(2)}$

Solution:

```
          0 1 0 1 1
      -----------------
1 1 1 | 0 1 0 0 1 1 0 1
        0 0 1 1 1
        ---------
            1 0 1 0
            0 1 1 1
            -------
              1 1 1
              1 1 1
              -----
              0 0 0
```

You can easily verify that $01001101_{(2)} = 77_{(10)}$, $111_{(2)} = 7_{(10)}$, and $01011_{(2)} = 11_{(10)}$

Example 2: Calculate $01111001_{(2)} \div 1011_{(2)}$

Solution:

```
          01011
     ___________
1011 | 01111001
     | 01011
_____|-
       ________
         10000
        -1011
       ________
          1011
          1011
         -
       ________
          0000
```

You can easily verify that $01111001_{(2)} = 121_{(10)}$, $1011_{(2)} = 11_{(10)}$, and $01011_{(2)} = 11_{(10)}$

5.6 Fixed Point and Floating Point Number Representations

5.6.1 Fixed Point Representation

To learn how computers represent real numbers using the fixed-point representation of numbers. We will first learn the concept using the decimal number system.

Suppose you have been told to write all real numbers by using the following rule.

The number will always have exactly four digits before the decimal point and exactly three digits after the decimal point.

Second column of the following table shows how different numbers will be written by using this rule.

Number	Number Written By Using the rule	Number without decimal point
73.4	0073.400	0073400
120.3456	0120.345 **(6 cannot be written)**	0120345
110	0110.000	0110000
11101.0	1101.0000 **(so cannot be represented)**	11010000

These numbers can be called fixed-point numbers because the position of the decimal point is fixed within the number. If numbers are written by using this format, we do not need to write the decimal point as it is always on the left of right digit from the left-hand side. This is shown in the third column of the table.

It is also clear from the table above that the numbers having more than 4-digit integeral part or more than 3-digit fractional part can not be represented accurately using this representation.

Computers are made to represent real numbers in a similar way. To represent a real number in a computer it may be programmed to use the following rules.

- Numbers may be represented using 8 bits, 16 bits, 32 bits or more.
- Decimal point will not be written.
- Decimal point is always after the 10^{th} bit.
- MSB is used to represent sign of the number (0 means +ve and 1 means –ve).
- Next 9 bits will be used to store the integral part of the number.
- Remaining 6 bits will be used to represent the fractional part of the number

This format is shown below:

Sign Bit	Integral part of the number	Fractional Part

This representation of real numbers is called the fixed-point representation. Following table shows how some of the binary numbers that can be represented using this representation.

Decimal Number	Binary Number	Number in Fixed-Point Form
3.625	011.1010	0000000011101000
247.90625	11110111.11101	0011110111111010
-7.66796875	-0111.10101011	1000000111101010 (remaining bits do not fit)
-81.765625	-1010001.110001	1001010001110001

Advantage of this representation is that it is very simple to use and the disadvantage is that very small or very large numbers cannot be represented using it.

Following examples show the process of representing different numbers in this form.

Example 1: Represent 23.6 in 16-bit fixed-point form using 10 bits for integral part.

Solution:

$23 = 010111_{(2)}$ and $0.6 = .1001001$

$23.6 = 010111.1001001 = 0000010111.100100$

and in fixed point form $23.6 = 0000010111100100$

Example 2: Represent -36.25 in 16-bit fixed-point form using 10 bits for integral part.

Solution:

$36 = 0100100_{(2)}$ and $0.25 = .01$

$36.25 = 0100100.01 = 0000100100.01$

And in fixed point form $-36.25 = 1000100100010000$

The following examples show the conversion of fixed-point numbers into decimals.

Example 3: Convert the 16-bit fixed-point number 0100010111100100 into decimal where 10 bits are used for integral part

Solution:

Integral part = 0100010111 and **Fractional Part** = .100100

Now $0100010111_{(2)} = 279$

$100100_{(2)} = 0.5 + .0625 = 0.5625$

So $0100010111100100 = 279.5625_{(10)}$

Example 4: Convert the 16-bit fixed-point number 1 0 0 0 1 1 0 1 1 1.1 1 0 0 0 0 into decimal where 10 bits are used for integer part.

Solution:

The Integral part = 1 0 0 0 1 1 0 1 1 1 and **Fractional Part** = 1 1 0 0 0 0

Now 1 0 0 0 1 1 0 1 1 1 $_{(2)}$ = -55

And . 1 1 0 0 0 0 $_{(2)}$ = 0.5 + .25 = 0.75

So 0 1 0 0 0 1 0 1 1 1 1 0 0 1 0 0 = -55.75 $_{(10)}$

5.6.2 Floating–Point Representation

Floating point representation is another useful way of writing real numbers. In this format, very small and very large numbers can be represented efficiently. Before learning the floating-point representation of binary numbers let us consider the following format of writing the decimal in the notation:

$$174.592 = 0.174592 \times 10^3.$$

This representation is known as the scientific notation. In this notation, 10 is the base, power of 10 is called the **exponent** and the **number** is called the **mantissa** so in the above number base is 10, mantissa is 0.174592 and the exponent is 3.

We can also write binary numbers in a similar way. For example, the number $1000.1101 = 0.10001101 \times 2^4$ where in this case the base is 2, the mantissa is 10001101 and the exponent is 4. Now consider the following way of writing the binary numbers.

Write the sign	Write the Exponent	Write the Mantissa

For example the table given below shows different binary numbers written in this format. Note that the binary point is adjusted such that leading bit in the mantissa of all numbers is always 1.

Number	Sign	Exponent	Mantissa
1.10001101×2^4	+	4	1.10001101
-1.1101101×2^5	-	5	1.1101101
$1101.0011 = 1.1010011 \times 2^3$	+	3	1.1010011
$0.011011 = 1.1011 \times 2^{-2}$	+	-2	1.1011

This format for writing real numbers is called the floating point representation. Most digital computers use this format for representing real numbers. In this book we will use the following floating-point format:

S	6-bit Exponent						9-bit Mantissa								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

So the computers use 16 bits for representing floating–point numbers. The MSB marked as **S** is used to represent the sign of the number. Next 6 bits are used to represent the

exponent and the remaining 9 bits are used to represent mantissa of the number. Following are some important points about this representation:

- The **sign** of a binary floating-point number is represented by a single bit. A 1 bit indicates a negative number, and a 0 bit indicates a positive number.
- The **exponent** is a signed integral and is represented as 6-bit 2's complement number.
- As first bit of the **mantissa** is always 1, so in most modern computers it is not written.

Following three examples show the process of representing numbers in floating–point format.

Example 1: Represent 17.5 as a 16-bit floating point number.

Solution:

Step 1: convert the number into binary

$$17.5 = 010001.10_{(2)} = 1.00110 \times 2^4$$

Step 2: Represent the number in floating–point format.

Sign = + = 0

Exponent = 4 and in 6-bit 2'scomplement form 000100

Mantissa = 1.00110 = 1.001100000

So the number in floating–point format is

S	6-bit Exponent						9-bit Mantissa								
0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0

Note : The first 1 and the binary point in the mantissa is not written.

Example 2: Represent -117.125 as a 16-bit floating point number.

Solution:

Step 1: convert the number into binary

$$-117.125 = -01110101.0010_{(2)} = -1.1101010010 \times 2^6$$

Step 2: represent the number in floating point format

Sign = - = 1

Exponent = 6 and in 6-bit 2's complement form 000110

Mantissa = 1.1101010010 = 1.110101001

So the number in floating point format is

S	6-bit Exponent						9-bit Mantissa								
1	0	0	0	1	1	0	1	1	0	1	0	1	0	0	1

Example 3: Represent– 0 . 0 0 0 1 1 0 1 0 0 1 0 0 1 $_{(2)}$ as a 16-bit floating point number.

Solution:

$-0.0001101001001 = -1.101001001 \times 2^{-4}$

Sign = - = 1

Exponent = -4 and in 6-bit 2's complement form 1 1 1 1 0 0

Mantissa = 1 . 1 0 1 0 0 1 0 0 1 = 1 . 1 0 1 0 0 1 0 0 1 0

So the number in floating point format is

S	6-bit Exponent						9-bit Mantissa								
1	1	1	1	1	0	0	1	0	1	0	0	1	0	0	1

Following examples show the process of converting a floating-point number into binary.

Example 4: Convert the following 16-bit floating point numbers into binary.

S	6-bit Exponent						9-bit Mantissa								
0	0	1	1	1	1	0	1	1	0	0	0	1	0	0	0

Solution:

S = 0 = +ve

Exponent = 0 1 1 1 1 0 = 30

Mantissa = **1** . 1 1 0 0 0 1 0 0 0

So the number is $\mathbf{1}.110001000 \times 2^{30}$

Note that the first 1 in the mantissa is written because it was ignored when the number was written in floating point form.

Example 5: Convert the following 16-bit floating point numbers into binary

S	6-bit Exponent						9-bit Mantissa								
1	1	0	0	1	1	1	0	1	1	1	0	1	1	0	1

Solution:

S = 1 = -ve

Exponent = 1 0 0 1 1 1 = - 0 1 1 0 0 1 = -25

Mantissa = 1 . 0 1 1 1 0 1 1 0 1

So the number is $-1.011101101 \times 2^{-25}$

5.7 Computer Code

We know that alphabetic data consists of only characters and alphanumeric data consists of characters and numbers digits. To represent this data in a computer system we assign a numeric code to each character of the alphabet. For example, we may assign the following codes to different characters e.g. A:65, B:66 etc. Therefore, we can represent both alphabetic and alphanumeric data in a computer system by using these codes.

5.7.1 ASCII (American Standard Code for Information Interchange)

ASCII is one such coding scheme published by ISO (International Standards Organization). It is a 7-bit coding scheme. The codes assigned to various characters are shown in the table below.

TABLE OF ASCII CODES

Code	Character	Code	Character	Code	Character	Code	Character
0		32	Space	64	@	96	`
1		33	!	65	A	97	a
2		34	"	66	B	98	b
3		35	#	67	C	99	c
4		36	$	68	D	100	d
5		37	%	69	E	101	e
6		38	&	70	F	102	f
7		39	'	71	G	103	g
8		40	(	72	H	104	h
9		41	)	73	I	105	i
10		42	*	74	J	106	j
11		43	+	75	K	107	k
12		44	,	76	L	108	l
13		45	-	77	M	109	m
14		46	.	78	N	110	n
15		47	/	79	O	111	o
16		48	0	80	P	112	p
17		49	1	81	Q	113	q
18		50	2	82	R	114	r
19		51	3	83	S	115	s
20		52	4	84	T	116	t
21		53	5	85	U	117	u
22		54	6	86	V	118	v
23		55	7	87	W	119	w
24		56	8	88	X	120	x
25		57	9	89	Y	121	y
26		58	:	90	Z	122	z
27		59	;	91	[	123	{
28		60	<	92	\	124	|
29		61	=	93	]	125	}
30		62	>	94	^	126	~
31		63	?	95	_	127	

Note: There is no character corresponding to the codes 0 – 31 .

Note that the character **A** and **a** have different codes in ASCII. Most computers also use 8-bit ASCII codes. In the 8-bit ASCII codes the remaining 128 codes are used to represent graphical and other special characters.

Following examples demonstrate the use of ASCII to represent different messages.

Example 1: Represent the following message 'Binary' using ASCII codes.

Solution: Using the ASCII table we see that

Character	Decimal Code	Binary Code
B	66	01000010
i	105	01101001
n	110	01101110
a	97	01100001
r	114	01110010
y	121	01111001

So we can write
"Binary" as 01000010 01101001 01101110 01100001 01110010 01111001

Example 2: Convert the following ASCII message into English

01010111 01101000 01100001 01110100 00111111

Solution: Using the ASCII table we see that

Binary Code	Decimal Code	Character
01010111	87	W
01101000	104	h
01100001	97	a
01110100	116	t
00111111	63	?

So,we can write
01010111 01101000 01100001 01110100 00111111 to represent the message **What?**

5.7.2 Binary Coded Decimal (BCD)

This coding scheme is used to represent numeric data. We know that decimal numbers system has ten different digits. To represent these digits we need a 4-bit code. In BCD the digits are assigned the following codes.

Table of BCD Codes

Digit	Code	Digit	Code	Digit	Code	Digit	Code	Digit	Code
0	0000	1	0001	2	0010	3	0011	4	0100
5	0101	6	0110	7	0111	8	1000	9	1001

Following example shows the representation of non-negative integers in BCD.

Example: Represent 9807 in BCD.

Solution:

We know that in BCD 9 = 1001 , 8 = 1000 , 0 = 0000 and 7 = 0111

Thus 9807 = 1001 1000 0000 0111

Clearly we need 16 bits to represent this 4-digit number. The same number can be represented in binary by using 14 bits. So the BCD codes use more bits hence require more

computer memory. When arithmetic is to be performed on the numbers coded in BCD either they are first converted into binary and then arithmetic is performed or special circuits are designed for this purpose.

5.7.3 Extended Binary Coded Decimal Interchange Code (EBCDIC)

IBM introduced a new way of character coding scheme called EBCDIC (Extended Binary Coded Decimal Interchange Code). It was a developed form of some of the existing codes like BCD. It is an 8-bit code so 256 different characters can be represented in EBCDIC. It was the most frequently used character code but with the increased use of the personal computer and computer networks the ASCII coding scheme became the standard coding scheme and now most of the computers use ASCII. Following table gives some of the characters and their EBCDIC codes.

Table showing EBCDIC Codes for different characters

Hex Code	Character	Hex Code	Character	Hex Code	Character	Hex Code	Character
C0	{	D0	}	E0	\	F0	0
C1	A	D1	J	E1		F1	1
C2	B	D2	K	E2	S	F2	2
C3	C	D3	L	E3	T	F3	3
C4	D	D4	M	E4	U	F4	4
C5	E	D5	N	E5	V	F5	5
C6	F	D6	O	E6	W	F6	6
C7	G	D7	P	E7	X	F7	7
C8	H	D8	Q	E8	Y	F8	8
C9	I	D9	R	E9	Z	F9	9

5.7.4 Unicode

Unicode is another popular coding scheme used these days. It is a 16-bit coding scheme so more than $2^{16} = 65536$ characters can be represented in this coding scheme

Exercise

1. Explain the following:
   a. Binary number system
   b. Octal number system
   c. Decimal number system
   d. Hexadecimal number system
   e. ASCII codes
   f. BCD

2. Explain the following terms with examples.
   a. Data
   b. Information
3. What are the main types of data used in different computer applications? Explain the uses of each of the data types and the operations performed on it.
4. Explain the 1's complement method of representing signed numbers. How can you perform subtraction using this method?
5. Explain the 2's complement method of representing signed numbers. How can you perform subtraction using this method?
6. Convert the following decimal numbers into binary, octal and hexadecimal
   a. 78
   b. 97
   c. 129
7. Convert the following hexadecimal numbers into binary, octal and decimal
   a. $7A_{(16)}$
   b. $1C2_{(16)}$
   c. $89_{(16)}$
8. Convert the following octal numbers into binary, decimal and hexadecimal
   a. $125_{(8)}$
   b. $57_{(8)}$
   c. $777_{(8)}$
9. Convert the following binary numbers into octal, decimal and hexadecimal
   a. $01110101_{(2)}$
   b. $10101001_{(2)}$
   c. $00110011_{(2)}$
10. Convert the following BCD numbers into Decimal00111001
    a. 00111001 b. 00000111
    c. 10000001
11. Represent the following numbers as 8-bit 1's complement and 10-bit 2's complement numbers
    a. 76
    b. -98
    c. -126
12. Represent the following 8-bit 1's complement numbers into decimal
    a. 00101011
    b. 10001001
    c. 11111111
13. Represent the following 8-bit 2's complement numbers into decimal
    a. 00111101
    b. 11111111
    c. 10101010
14. Perform the following subtraction using 8-bit 1's complement method. Verify your answer by converting it into decimal. All numbers are in decimal system.
    a. 57 – 126
    b. 120 – 76
    c. -20 - 52
15. Perform the following subtraction using 8-bit 2's complement method. Verify your answer by converting it into decimal. All numbers are in decimal system.
    a. 127 – 126
    b. 12 – 106
    c. -12 – 25

16. Perform the following subtraction using 8-bit 1's and 2's complement method. Verify the results by converting your answer into decimal. Explain why the results are not correct if there is needed.
a. -57 – 96
b. -120 – 110
c. - 60- 68

17. Perform the following subtraction using 10-bit 1's and 2's complement method. Verify the results by converting your answer into decimal.
a. -57 – 96
b. -120 – 110
c. - 60- 68

18. What are the smallest and largest numbers that could be represented in 8 bits?

19. What are the smallest and largest numbers that could be represented in 8 bit 1's complement form?

20. What are the smallest and largest numbers that could be represented in 8 bit 2's complement form?

21. Represent the following numbers using fixed point representation. Use the following format for the conversion. Also verify your results by converting your results back into decimal.
a. 25.5
b. 233.9
c. 33.6

10 bits for integral part	6 bits for fractional part

22. Represent the following numbers using fixed point representation. Use the format given in the previous question for the conversion. Explain if there is any trouble.
a. 1025.5
b. -1233.9
c. -2333.6

23. Represent the following numbers using floating point representation.
Use the floating point format given in the chapter
a. 1025.5
b. -1233.9
c. -2333.6

24. Represent the following messages using the ASCII codes given in the table of ASCII codes. Also verify your coded message by converting it back into English. (Do not forget to convert the space character)
a. He is a good student
b. 2 + 2 = 4
c. I like Computer Science
d. Binary numbers are GREAT

25. Fill in the blanks:
(i) ______________ is raw facts of knowledge which is ready for processing.
(ii) Processed data is called ______________.
(iii) ______________, ______________, and ______________ are three methods of representing signed numbers.
(iv) ASCII stands for ______________.
(v) 1 0000 0100 0010 is equal to ______________$_{(16)}$.
(vi) 1 000 100 010 is equal to ______________$_{(8)}$.
(vii) 2's complement of $0010\ 0011_{(2)}$ is ______________.
(viii) Computer manipulates everything in terms of ______________.

(ix) Base of hexadecimal numbers is ________.
(x) In ________ end carry is discarded.

26. Match the following:

Data	American Standard Code for Information Interchange
Processing	$22_{(10)}$
Information	Processed Data
ASCII	Raw facts to which no meaning is attached and is ready for processing
$16_{(16)}$	Processing means to manipulate, calculate, distribute or arrange
$12_{(16)}$	$22_{(8)}$

27. Choose the correct answer:

a. The hexadecimal number $10_{(16)}$ is equal to
(i) $10_{(10)}$ (ii) $100_{(10)}$ (iii) $16_{(10)}$ (iv) All of above

b. The hexadecimal number $100_{(16)}$ is equal to
(i) $0001\ 0000\ 0000_{(2)}$ (ii) $256_{(10)}$ (iii) $400_{(8)}$ (iv) All of above

c. 2's complement of $0101010_{(2)}$ is
(i) 1010110 (ii) 1010101 (iii) 0000011 (iv) None of above

d. 1's complement of a negative binary number can be calculated by
(i) reversing the bits in the number
(ii) reversing the bits in the number and adding one
(iii) can not be calculated (iv) both (i)and (ii)

e. (011)4752105 is
(i) numeric data (ii) alphanumeric data (iii) alphabetic data
(iv) both(b)and(c)

28. Mark the following as True/False:

a. It is impossible to build a computer that uses decimal number system
b. $1234_{(16)} = 110\ 11\ 100_{(2)}$ **c.** All computers in the world use ASCII.
d. 1's and 2's complement methods works only for fixed numbers of bits
e. We can not represent 256 using 8 Bits
f. There are total of 8 basic digits in octal number system
g. ASCII is a 7-bit coding scheme
h. Unicode is used to provide multilingual support in software
i. BCD stands for Binary Coded Digits
j. The value of G represents 16 in hexadecimal number system.

Answers

Q.25
(i) Data (ii) Information (iii) signed magnitude, 1's Complement, 2's Complement
(iv) American Standard Code for Information Interchange (v) 1024 (vi) 1024
(vii) 1101 1101 (viii) Binary numbers (ix) 16 (x) 2's complement

Q.27
a. (iii) b. (iv) c. (i) d. (i) e. (ii)

Q.28
a. F b. F c. F d. T e. T f. T g. T h. T i. F j. F

Chapter 6

Boolean Algebra

6.1 Introduction

Boolean algebra is the algebra of logic. It uses symbols to represent logical statements instead of words. Boolean algebra was formulated by the English Mathematician George Boole in 1847. Boolean algebra consists of rules for manipulating symbols. Boolean algebra has exactly the same structure as propositional calculus. The most important application of Boolean algebra is in digital logic. Computer chips are made up of transistors that are arranged in logical gates. Each gate performs a simple logical operation. The computer processes the logical propositions in its program by processing electrical pulses. The design of a particular circuit is based on a set of logical statements. These statements can be translated into the symbols of Boolean algebra. The algebraic statements can then be simplified according to the rules of the algebra, and translated into a simpler circuit design. Boolean algebra return results in terms of true or false i.e. 1 or 0 respectively.

Consider the following statements:

I am Pakistani	1
2 + 2 = 5	0
Lahore is the capital of Pakistan	0
5 + 1 = 6	1

Each of these statements is either **TRUE** or **FALSE**. Such statements are called **propositions**. The sentence **What is your name?** is not a proposition because it has no truth-value (TRUE or FALSE).

We can combine two propositions to form a new proposition as follows:

Let **p = Islamabad is the capital of Pakistan**
and **q = Sialkot is the capital of Punjab.**

Then **p** is **TRUE** and **q** is **FALSE**

Now form a new proposition **t** by using p and q as follows:

t = (Islamabad is the capital of Pakistan) **AND** (Sialkot is the capital of Punjab) or we may write that

t = p AND q

This proposition is **FALSE** because q is False and for **t** to be TRUE both p and q must be **TRUE**.

Similarly let **r = p OR q**

Clearly the proposition **r** is **TRUE** because **p** is **TRUE**.

Also with every proposition p we can make another proposition q as follows

Let **p = Islamabad is the capital of Pakistan**

Then make a new proposition **q** as follows:

q = NOT (Islamabad is the capital of Pakistan)

We may write **q = It is not TRUE that** Islamabad is the capital of Pakistan

q is called the negation of **p** and we write **q = NOT (p)** to express this idea.

It is obvious from the definition of negation given above that if p is TRUE then NOT (p) will be FALSE and if **p** is FALSE then NOT(p) will be TRUE.

Thus we have the following key points:

- Each proposition is either **TRUE** or **FALSE**
- We have two ways (AND ,OR) of combing two propositions to make new propositions
- Each proposition p has a negation NOT (p).

George Boole was actually interested in representing such a system of Logical sentences in a mathematical form.

Now let us consider another system, we know that all electronic devices consist of circuits of switches (Transistors). A switch at any given time is in one of the two states **ON** or **OFF**.

We can combine two switches A and B in the following two ways:

Series: The two switches A and B are arranged in a series as shown in the figure 6.1 the bulb will be **ON** if both switches are **ON** and it will be **OFF** otherwise.

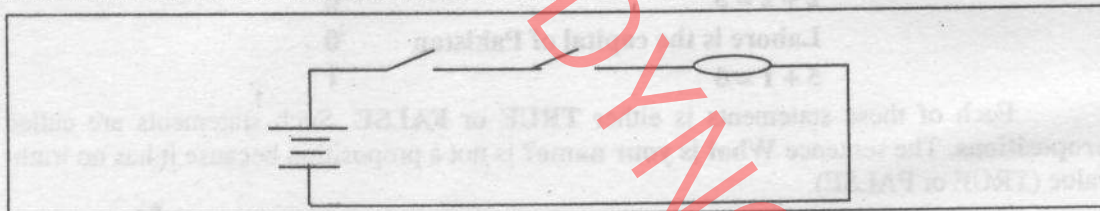



Figure 6.1

Parallel: If A and B are arranged in parallel as shown in the figure 6.2.

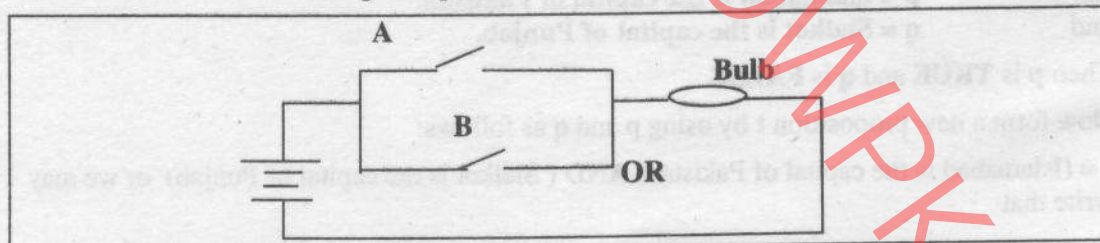



Figure 6.2

The bulb will be **ON** if atleast one of the switches is **ON** otherwise it will be **OFF**.

Serial circuit is represented by **.** operator and parallel circuit by + operator. This is explained as under:

Operations of AND (.)		
Switch A	Switch B	BULB
OFF	OFF	OFF
OFF	ON	OFF
ON	OFF	OFF
ON	ON	ON

Serial Circuit

Operations of OR (+)		
Switch A	Switch B	BULB
OFF	OFF	OFF
OFF	ON	ON
ON	OFF	ON
ON	ON	ON

Parallel Circuit

We can also represent the above circuits as expressions of the form **A. B** read as A dot B and **A+B** read as A plus B.

6.2 Boolean Algebra

Two valued Boolean algebra is a set that has two elements and two operations usually denoted by **.** and **+** are defined on the set such that the following axioms are satisfied

Close: Set *B* is closed under **.** and **+**

Commutative: Both the operations are commutative which means that if a and b are two variables that take values from the set, then

$$a + b = b + a \quad \text{and} \quad a \,.\, b = b \,.\, a$$

Associative: Both the operations are associative which means that if a, b and c are variables that take values from the set B, then

$$a+(b + c) = (a + b)+c \quad \text{and} \quad a.(b. c) = (a . b).c$$

Distributive: The **.** operation is distributive over **+** and **+** operation is distributive over **.** operation so if a, b and c are variables that take values from the set, then

$$a.\,(b + c) = (a \,.\, b) + (a \,.\, c) \quad \text{and} \quad a + (b. c) = (a + b).(a + c)$$

Identity: There is an identity element 1 with respect to **.** and an identity element 0 with respect to + such that for all x,

$$x.\,1 = x \text{ and } x + 0 = x$$

Complement: Each element of the set B has a complement. If x is an element of set then its complement is denoted by $\bar{x}$ and has the following properties:

$$x + \bar{x} = 1 \quad \text{and} \quad x \,.\, \bar{x} = 0$$

Following example defines the most commonly used two valued Boolean algebra.

Example: Consider the set B = {0, 1} and two operations + and . on B as follows:

Operations of AND (.)		
x	y	$x . y$
0	0	0
0	1	0
1	0	0
1	1	1

Operations of OR (+)		
x	y	$x + y$
0	0	0
0	1	1
1	0	1
1	1	1

This set is a Boolean Algebra.

Note that this addition operation is different from the usual addition because $1 + 1 = 1$. Both the operations . and + have the following properties.

If x, y and z are variables that take values from the set B-then the set is closed under e + operator.

Close: Both $x \cdot y$ and $x + y$ are members of B (i.e. $x \cdot y$ and $x + y$ are either 0 or 1). Thus set B is closed under the operations . and +

Commutative: Following tables show that the operations of . and + commutative because from the tables it is obvious that

$$x \cdot y = y \cdot x \quad \text{and} \quad x + y = y + x$$

Tables to show that . and + are commutative.

x	y	$x \cdot y$	$y \cdot x$
0	0	0.0 = 0	0.0 = 0
0	1	0.1 = 0	1.0 = 0
1	0	1.0 = 0	0.1 = 0
1	1	1.1 = 1	1.1 = 1

x	y	$x + y$	$y + x$
0	0	0+0 = 0	0+0 = 0
0	1	0+1 = 1	1+0 = 1
1	0	1+0 = 1	0+1 = 1
1	1	1+1 = 1	1+1 = 1

Associative: Following table shows that for all values of the Boolean variables x, y and z we have $x \cdot (y \cdot z) = (x \cdot y) \cdot z$. so the AND operation is associative. Similarly we can show that OR operation is also associative i.e. $x + (y + z) = (x + y) + z$.

Table to show that . is associative

x	y	z	$x \cdot y$	$(x \cdot y) \cdot z$	$y \cdot z$	$x \cdot (y \cdot z)$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

Distributive:

As is shown in the table below that for all possible values of x, y and z, we have

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

so is distributive over +. Similarly we can show that $x + (y \cdot z) = (x + y) \cdot (x + z)$

x	y	z	$x \cdot y$	$x \cdot z$	$x \cdot y + x \cdot z$	$y + z$	$x \cdot (y + z)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	1	0	0	0	1	0
1	0	0	0	0	0	0	0
1	0	1	0	1	1	1	1
1	1	0	1	0	1	1	1
1	1	1	1	1	1	1	1

Identity

As is obvious from the table given below that for any value of the variable x

$$0 + x = x \quad \text{and} \quad x \, . \, 1 = x$$

so 0 is the identity element of + and 1 is the identity element of .

x	$x.1$	$x+0$
0	0	0
1	1	1

Complement

Every element of B has a complement. Complement of 0 is 1 and complement of 1 is 0 because

$$0 + 1 = 1 \quad \text{and} \quad 0 \, . 1 = 0$$

So the set B ={0,1} with the defined operations is a Boolean algebra because it satisfies all the axioms of Boolean Algebra.

Boolean Constants

If B = [0, 1] with operations . and + is a Boolean Algebra, then 0 and 1 are called Boolean constants.

What are the Boolean constants in the Boolean algebra given in the examples?
What values can the Boolean variables take in Boolean Algebra?

Boolean Variables

If B = {0, 1} with operations . and + is a Boolean algebra, then the variables x, y etc are called Boolean variables. We can use the Boolean constants and variables to form Boolean expressions.

Boolean Expressions

If x, y and z are Boolean variables and 0 and 1 are the Boolean constants, then by using the . , + and complement operations we can combine two or more variables and constants to make comparision.

$$x + y. \, z \quad \text{and} \quad \overline{x} \, . \, (y + z) \text{ etc.}$$

For evaluating a Boolean expression, we follow the following precedence of operations :

1. First of all evaluate all the complement operations
2. Secondly evaluate all the product .
3. Evaluate the addition operations + at the end

We can use parentheses to change the order of evaluation of operations in a Boolean expression. If parentheses are used, then first of all that part of expression is evaluated which is within the parentheses.

Following example use these rules to evaluate different Boolean expressions:

Example 1: Evaluate $\overline{x}.y + x.\overline{z} + x.\overline{y}$ for $x = 0, y=1$ and $z = 0$

Solution:

First calculate complements as $x = 0$ so $\overline{x} = 1$ similarly $\overline{y} = 0$ and $\overline{z} = 1$

Now calculate products so $\overline{x}.y = 1.1 = 1$ $\quad x.\overline{z} = 0.1 = 0$ and $x.\overline{y} = 0.1 = 0$

So $\overline{x}.y + x.\overline{z} + x.\overline{y} = 1+0+0 = 1$

Example 2: Evaluate $(x + y).\ \overline{x} + (\overline{y} + z)$ for $x = 0, y = 1$ and $z=1$

Solution:

First of all calculate comlements

$x + y = 0 + 1 = 1$ similarly $\overline{y} + z = 0 + 1$

Now $\overline{x} = 1,\ \overline{y} = 0,\ \overline{z} = 0$

So $(x + y).\ \overline{x} + (\overline{y} + z) = (0 + 1).1 + (0 + 1) = 1.1 + 1 = 1 + 1 = 1$

6.2.1 Evaluating an expression for all possible input values

Following examples shows the use of truth table for evaluating an expression for all possible input values.

Example 1: Evaluate the following Boolean expression. $x.\overline{y} + \overline{x}.y$ using a truth table.

Solution:

x	$\overline{x}$	y	$\overline{y}$	$x.\overline{y}$	$\overline{x}.y$	$x.\overline{y} + \overline{x}.y$
0	1	0	1	0	0	$0+0=0$
0	1	1	0	0	1	$0+1=1$
1	0	0	1	1	0	$1+0=1$
1	0	1	0	0	0	$0+0=0$

Example 2: Evaluate the following Boolean expression. $x.y + \overline{x}.y + y.\overline{z}$ using a truth table.

Solution:

x	$\overline{x}$	y	$\overline{y}$	z	$\overline{z}$	$x.y$	$\overline{x}.y$	$y.\overline{z}$	$x.y + \overline{x}.y + y.\overline{z}$
0	1	0	1	0	1	0	0	0	0
0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	0	1	0	1
0	1	1	0	1	0	0	1	0	1
1	0	0	1	0	1	0	0	0	0
1	0	0	1	1	0	0	0	1	1
1	0	1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0	0	1

It is often very useful to construct a truth table of a given Boolean expression. It is important to note that truth table of a two variable expression will always have $2^2 = 4$ rows and truth table of a 3 variable expression will always have $2^3 = 8$ rows

6.2.2 Boolean Functions

Consider the Boolean expression $x + y$ where x and y are Boolean variables. Now let a function f as follows:

- f takes two Boolean constants as input
- f then calculate the value of the above expression at the input values
- The calculated value is the final answer of f.

Examples of two valued functions are:

$f(x, y) = x + y$ and $g(x, y) = \bar{x} . \bar{y} + x.y$

where x, y are Boolean variables.

Now consider another Boolean expression $x + y. z$, where x, y and z are Boolean variables. Now let us make the following rule for calculating the value of g as follows:

- g takes two Boolean constants as input
- g then calculate the value of the above expression at the input values
- The calculated value is the final answer of g.

Example: Represent the function $\mathbf{f}(x, y) = x. \bar{y} + \bar{x}.y$ by using a truth table

Solution:

x	y	$\bar{x}$	$\bar{y}$	$x.\bar{y}$	$\bar{x}.y$	$\mathbf{f}(x, y) = x.\bar{y} + \bar{x}.y$
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	1	0	0	0	0	0

This truth table shows the value of the functions for all the possible values of the parameters.

6.3 Laws and Theorems of Boolean Algebra

In this section we will see different laws of Boolean algebra and also prove some useful theorems. These theorems are used for simplifying different Boolean functions and in the simplification of different logical circuits

Theorem 1: If x is a Boolean variables then $x . x = x$ and $x + x = x$.This is also known as the idempotent law

We can prove this theorem in the following two ways:

- By using a truth table
- By using the axioms of Boolean algebra

Proof: By using a truth table

x	$x \cdot x$
0	$0 \cdot 0 = 0$
1	$1 \cdot 1 = 1$

From the truth table given above it is clear that if $x = 0$ then $x + x$ is also 0 and if $x = 1$ then $x + x$ is also 1 so we can say that $x + x = x$

Note: All theorems of Boolean algebra can be proved by using truth tables.

by using the axioms of Boolean algebra

Now we shall prove the second part of this theorem by using the axioms of Boolean algebra and the definition of . and +as follows:

$$\text{L.H.S} = x + x$$
$= x.1 + x.1$ (by identity element)
$= x.(1+1)$ (by distributive law)
$= x.1$ $(1 + 1 = 1)$
$= x$ (by identity element)
$=$ R.H.S

Hence the theorem is proved.

Note: The second part can be obtained by changing . into +
This fact will be very useful for proving certain theorems

Theorem 2: If x is a Boolean variable then $x + 1 = 1$ and $x \cdot 0 = 0$

We can prove this theorem using a truth table but that is left as an exercise for you. Here we shall prove this theorem by using axioms of Boolean algebra and previously proved theorems

Proof: L.H.S $= x + 1$
$= x + (x + \bar{x})$ (by definition of complement)
$= (x + x) + \bar{x}$ (by associative law)
$= x + \bar{x}$ (by idem potent law)
$= 1$ (by definition of complement)
$=$ R.H.S

Note that: We can use existing theorems to prove more theorems.

Now we shall prove the second part of this theorem that states $x.0 = 0$

L.H.S $= x \cdot 0$
$= x \cdot (x \cdot \bar{x})$ (by definition of complement)
$= (x \cdot x) \cdot \bar{x}$ (by associative law)
$= x \cdot \bar{x}$ (by idempotent law)
$= 0$ (by definition of complement)
$=$R.H.S

Hence the theorem is proved.

Theorem 3: **For any Boolean variable x, $\overline{\overline{x}} = x$. This is also known as involution (or cancellation property).**

Proof: As we stated earlier that every theorem can be proved by using a truth table. Here we will use a truth tale to prove this theorem

x	$\overline{x}$	$\overline{\overline{x}} = x$
0	1	0
1	0	1

The result can be obtained by comparing the first and third column of the truth table

Theorem 4: **If x and y are Boolean variables then $x + x.y = x + y$ and $x.(x + y) = x$ this result is also known as the absorption law.**

Proof: L.H.S $= x + x.y$

$= x.1 + x.y$ (1 is identity element)

$= x.(1 + y)$ (distributive law)

$= x(1)$ $(1 + y = 1)$

$= x$ (1 is the identity element)

= R.H.S

The proof of the second part is similar and is left for the students as an exercise.

Hence the result follows.

Theorem 5

De Morgan's law: The complement of addition of two numbers is equal to the product of their complements. Similarly the complement of product of two numbers is equal to the sum to their complements.

If x and y are two Boolean variables then

$$\overline{x + y} = \overline{x}.\overline{y} \quad \text{and} \quad \overline{x.y} = \overline{x} + \overline{y}$$

Proof: We will prove the first result of this theorem by using truth table

x	y	$\overline{x}$	$\overline{y}$	$x + y$	$\overline{x + y}$	$\overline{x}.\overline{y}$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

From the last two columns of this table it is obvious that $\overline{x + y} = \overline{x}.\overline{y}$

6.3.1 Duality Principle

The **Principle of Duality** states that any result deduced from the axioms of Boolean algebra remains valid if the following steps are performed

- All 0's in the result are changed to 1 and vice versa
- The . in the original result is changed to + and vice versa

Note:	This result is very important because if we can prove a result of Boolean algebra then another valid result can be directly obtained from the proved result.

Example1: Prove that $\overline{x.y} = \bar{x} + \bar{y}$

Proof: We know from theorem 5 that $\overline{x + y} = \bar{x} . \bar{y}$ now applying the principle of duality on $\overline{x+y} = \bar{x}.\bar{y}$ gives us $\overline{x.y} = \bar{x} + \bar{y}$

Hence the result proved.

Example2: Apply the principle of duality to get the dual of the following expressions

$x.x = x$, $x + 1 = 1$, $x + \bar{x}.y = x + y$, $\overline{x + y} = \bar{x}.\bar{y}$ and

$x.(y + z) = (x.y) + (x.z)$

Solution:

i. By changing the only . to + we get $x + x = x$

ii. By changing the + to . and changing 1 to 0 gives $x . 0 = 0$

iii. By changing + to . and . to + gives $x . \bar{x} + y = x.y$

iv. Changing + to . and . to + gives us $\overline{x.y} = \bar{x} + \bar{y}$

v. Changing + to . and . to + gives us $x + (y.z) = (x + y).(x + z)$

6.3.2 Simplifying a Boolean function

It is clear from the above examples that every Boolean function can be represented as a combination of Boolean functions and also every circuit of logical gates can be represented as a Boolean expression. As the internal architecture of the computers memory and processor consists of these gates so it is always useful to find a simpler expression for representing a function. A simpler expression results into simple and efficient hardware.

In this section we shall learn the process of simplifying a given Boolean function. We will learn two ways of simplifying a Boolean function.

- Simplifying a Boolean function by using laws of Boolean algebra
- Simplifying a Boolean function by using k-map algorithm

The process of simplifying a Boolean function using laws of Boolean algebra are demonstrated in the following examples

Example 1: Simplify the following Boolean function:

$$f(x, y) = x + \bar{x}.y$$

Solution:

$f(x, y) = x + \bar{x}.y$

$= (x + \bar{x}).(x + y)$ (by Distributive law)

$= 1.(x + y)$ (by Complement Definition)

$= (x + y)$ (by definition of identity element)

Clearly to implement the non-simplified function needs three logic gates whereas the implementation of the simplified function needs only 1 logic gate.

Example 2: Simplify the following Boolean function f :

$$f(x, y, z) = \bar{x}.y.z + x.\bar{y} + \bar{x}.\bar{y}.z$$

Solution:

$f(x, y, z) = \bar{x}.y.z + x.\bar{y} + \bar{x}.\bar{y}.z$

$= \bar{x}.y.z + \bar{x}.\bar{y}.z + x.\bar{y}$ (By commutative law)

$= \bar{x}.z(y + \bar{y}) + x.\bar{y}$ (By distributive law)

$= \bar{x}.z.1 + x.\bar{y}$ (By definition of Complements)

$= \bar{x}.z + x.\bar{y}$ (Identity element)

It is obvious that to implement the non-simplified function needs 9 logic gates whereas the implementation of the simplified function needs only 5 logic gates

Example 3: Simplify the following Boolean function f :

$$f(x, y, z) = x.z + \bar{x}.z.y$$

Solution:

$f(x, y, z) = x.z + \bar{x}.z.y$

$= x.z + \bar{x}.y.z$ (by associative and commutative law)

$= (x + \bar{x}.y).z$ (by Distributive law)

$= (x + y).z$ (by Idempotent law)

$= x.z + y.z$ (by Distributive law)

Clearly the simplified functions are much more desirable and useful.

6.3.3 Disadvantages of using Boolean Algebraic laws

Following is the list of disadvantages of using Boolean algebraic laws for simplification of Boolean expressions:

- It is very difficult to write a computer program (automate) that can use these laws to simplify a given Boolean function.
- This process may not give the best-simplified function and different people can have different simplified expressions.
- For this process to work a Boolean function is needed but in most engineering applications we do not have the actual Boolean function but have its truth table of the required function.

To overcome these disadvantages Maurice Karnaugh established another method for simplifying a Boolean expression. This method is based upon the Boolean algebraic laws but has non-of these disadvantages. This is commonly known as the k-map method of simplification.

Before we learn the next method of simplification let us learn the following terms.

Literals: If we have a Boolean function of two variables x and y then each variable can appear in the function in two forms i.e either the variable itself appears or it appears in the complement form. Each of these forms is called a literal. Each literal represent on input to the Boolean function.

6.3.4 Minterms (Standard Product)

If we have a two Boolean variables x and y then we can form the following four products using these variables. $x \cdot y$, $x \cdot \overline{y}$, $\overline{x} \cdot y$, $\overline{x} \cdot \overline{y}$. These are called standard products or minterms with two variables. Following example list all the minterms with three variables.

Example: List down all the minterms of three variables x, y, z. Also give a general formula for calculating the number of minterms with n variables.

Solution: With three variables we can form the following minterms

$$x \cdot y \cdot z \qquad x \cdot y \cdot \overline{z}, \qquad x \cdot \overline{y} \cdot z, \qquad x \cdot \overline{y} \cdot \overline{z},$$
$$\overline{x} \cdot y \cdot z, \qquad \overline{x} \cdot y \cdot \overline{z}, \qquad \overline{x} \cdot \overline{y} \cdot z, \qquad \overline{x} \cdot \overline{y} \cdot \overline{z},$$

we can construct $4 = 2^2$ minterms with 2 variables and $8 = 2^3$ minterms with 3 variables. It is easy to see that we will have 2^n minterms with n variables. The table given below shows the way we name these minterms. It is important to remember the value of variables which is associated with a minterm.

Name	x	y	z	Minterm
m0	0	0	0	$\overline{x} \cdot \overline{y} \cdot \overline{z}$
m1	0	0	1	$\overline{x} \cdot \overline{y} \cdot z$
m2	0	1	0	$\overline{x} \cdot y \cdot \overline{z}$
m3	0	1	1	$\overline{x} \cdot y \cdot z$
m4	1	0	0	$x \cdot \overline{y} \cdot \overline{z}$
m5	1	0	1	$x \cdot \overline{y} \cdot z$
m6	1	1	0	$x \cdot y \cdot \overline{z}$
m7	1	1	1	$x \cdot y \cdot z$

Table of Names of minterms

6.3.5 Maxterms (Standard Sum)

If we have a two Boolean variables x and y, then we can form the following four sums using these variables. $x + y$, $x + \overline{y}$, $\overline{x} + y$, $\overline{x} + \overline{y}$. These are called a standard sums or maxterms with two variables. It is easy to see that we will have 2^n maxterms with n Boolean

variables. The table given below shows the way we name these Maxterms.

Name	x	y	z	Maxterms
M0	0	0	0	$x+y+z$
M1	0	0	1	$x+y+\overline{z}$
M2	0	1	0	$x+\overline{y}+z$
M3	0	1	1	$x+\overline{y}+\overline{z}$
M4	1	0	0	$\overline{x}+y+z$
M5	1	0	1	$\overline{x}+y+\overline{z}$
M6	1	1	0	$\overline{x}+\overline{y}+z$
M7	1	1	1	$\overline{x}+\overline{y}+\overline{z}$

Table of Names of Maxterms

The concept of minters and maxterms is very useful for simplifying a Boolean function to a minimum number of literals.

Another important idea is that we can write every Boolean function as Sum of minterms or as Product of Maxterms. We will learn the Minterms concept in detail and leave the Maxterms for the next classes.

6.4 Karnaugh Map (K-Map)

Karnaugh Map is a very efficient way of solving Boolean functions. In this section we will learn to solve a two and three variables Boolean function in the form of a map.

6.4.1 Map for a two variable Boolean function

Following figure shows the arrangement of a two variable Boolean function in the form of a map so the square in row 0 and column 0 is mo and for the minterm the square at row 0 column 1 is m1.

x \ y	0	1
0	m0	m1
1	m2	m3

Let us Consider the function as sum of minterms as follows $f(x,y,z) = \overline{x} \cdot y + x \cdot \overline{y}$. This function can be written in a k-map as follows

x \ y	$\overline{y}$	y
$\overline{x}$	0	1
x	0	0

So to express a function in the form of a k-map we determine the minterms in that function and then write 1 in all those squares which correspond to a minterm present in the function and write 0 in the remaining squares.

6.4.2 Map for a three variable Boolean function

The map for representing a three variable function is shown below:

		$\overline{y}.\overline{z}$	$\overline{y}.z$	$y.z$	$y.\overline{z}$
	x\y.z	0.0	0.1	1.1	1.0
$\overline{x}$	0	m0	m1	m3	m2
x	1	m4	m5	m7	m6

It is extremely important to arrange the rows and columns as given in the above table. The process of representing a three valued function in a k-map is the same as for the two variable functions. Following examples show the process of representing a Boolean function in the form of k-map.

Example 1: Represent the following Boolean function in a three variable k-map

$$f(x, y, z) = x . y . \bar{z} + \bar{x} . \bar{y} . \bar{z} + x . \bar{y} . \bar{z} + \bar{x} . y . z$$

Solution:

Step 1: First represent the function as sum of minterms form.

$$f(x, y, z) = x . y . \bar{z} + \bar{x} . \bar{y} . \bar{z} + x . \bar{y} . \bar{z} + \bar{x} . y . z$$

This function is already in the required form

Step 2: For each minterm present in the function Mark a 1 in the corresponding square in the map and mark a 0 in all other squares

x\ yz	0.0 $\bar{y}.\bar{z}$	0.1 $\bar{y}.z$	1.1 $y.z$	1.0 $y.\bar{z}$
0 $\bar{x}$	1	0	1	0
1 x	1	0	0	1

Example 2: Represent the following Boolean function in a two variable k-map

$$f(x, y) = y$$

Solution:

Step 1: First represent the function as sum of minterms form

$$f(x, y) = y$$
$$= (x + \bar{x}) . y$$
$$= x . y + \bar{x} . y$$

Step 2: For each minterm present in the function Mark a 1 in the corresponding square in the map

x\y	0 $\bar{y}$	1 y
0 $\bar{x}$	0	1
1 x	0	1

6.4.3 Simplifying a Boolean Function of Two Variables Using k-map

Following examples show the process of simplification of a two variable Boolean function using a k-map.

Example 1: Simplify the Boolean function $f(x, y) = x.y + \bar{x}.y$

Solution:

Step 1: Represent the function in the form of a k-map. This is shown below:

$x \backslash y$	$\bar{y}$	y
$\bar{x}$	0	1
x	0	1

Step 2: Mark any groups of two or four adjacent 1 as shown below

$x \backslash y$	$\bar{y}$	y
$\bar{x}$	0	1
x	0	1

Step 3: Write simplified expression for each group.

The grouped minterms are $\bar{x}\ .\ y$ and $x\ .\ y$ as the value of x changes so we can write the following expression for this group of minterms

$$\bar{x}\ .\ y\ +\ x\ .\ y = y$$

Step 4: Write the final simplified form as a sum of products

$$f(x, y) = y$$

Example 2: Simplify the Boolean function $f(x, y) = x.\ \bar{y}\ +\ \bar{x}.y + x\ .\ y$

Solution:

Step 1: Represent the function in the form of a k-map. This is shown below

$x \backslash y$	0	1
$\bar{x}$ 0	0	1
x 1	1	1

Step 2: Mark any groups of two or four adjacent 1 as shown below.

$x \backslash y$	0	1
$\bar{x}$ 0	0	1
x 1	1	1

Step 3: Write simplified expression for each group.

The grouped minterms are $\bar{x}\ .\ y$ and $x\ .\ y$ and another group of minterms is $x\ .\ \bar{y}$ and $x\ .y$. As the value of x changes in the first group and value of y changes in the second group.

so expression for the first group = y

so expression for the second group = x

Step 4: Write the final simplified form as a sum of products

$$f(x, y) = x + y.$$

Example 3: Simplify the Boolean function

$$f(x, y) = x.\bar{y} + \bar{x}.y + x.y + \bar{x}.\bar{y}$$

Solution:

Step 1: Represent the function in the form of a k-map. This is shown below

$x \backslash y$	$\bar{y}0$	$y1$
0	1	1
1	1	1

Step 2: Mark any groups of two or four adjacent 1 as shown below:

$x \backslash y$	0	1
$\bar{x}$ **0**	1	1
x **1**	1	1

Step 3: Write simplified expression for each group. All the elements are 1 and there is only one group .

Step 4: Write the final simplified form as a sum of products in this case we write, $f(x, y) = 1$ because it is always 1

Example 4: Simplify the Boolean function $f(x, y) = x.\bar{y} + \bar{x}.y$

Solution:

Step 1: Represent the function in the form of a k-map. This is shown below

$x \backslash y$	$\bar{y}0$	$y1$
0	0	1
1	1	0

Step 2: Mark any groups of two or four adjacent 1 as shown below :

$x \backslash y$	$\bar{y}0$	$y1$
$\bar{x}$ **0**	0	1
x **1**	1	0

Note that the elements along the diagonal are not adjacent to each other

Step 3: Write simplified expression for each group

as there are no groups so we write the minterm corresponding to the each 1 in the map

$$x.\bar{y} \text{ and } \bar{x}.y$$

Step 4: Write the final simplified form as a sum of products

$$f(x, y) = x.\bar{y} + \bar{x}.y$$

6.4.4 Simplifying a Boolean Function of Three Variables Using k-map

Following examples show the process of simplification of a three variable Boolean function using k-map.

Example 1: Simplify the Boolean function

$$f(x, y, z) = x.y.\bar{z} + \bar{x}.\bar{y}.\bar{z} + x.\bar{y}.\bar{z} + \bar{x}.y.z$$

Solution:

Step 1: Represent the function in the form of a k-map. This is shown below

$x \backslash y.z$	$\bar{y}.\bar{z}$	$\bar{y}.z$	$y.z$	$y.\bar{z}$
$\bar{x}$	1	0	1	0
x	1	0	0	1

Step 2: Mark any groups of two or four adjacent 1 as shown below

$x\backslash y.z$	$\bar{y}.\bar{z}$	$\bar{y}.z$	$y.z$	$y.\bar{z}$
$\bar{x}$	1	0	1	0
x	1	0	0	1

Group 1: $\bar{x}.\bar{y}.\bar{z}$ and $x.\bar{y}.\bar{z}$ first row

Group 2: $x.\bar{y}.\bar{z}$ and $x.y.\bar{z}$ so third column

Ungrouped terms: $\bar{x}.y.z$

Step 3: Write simplified expression for each group.
as there are two groups so we write the minterm corresponding to the each 1 in the map

Group 1: $\bar{x}.\bar{y}.\bar{z}$ and $x.\bar{y}.\bar{z}$ so simplified expression is $\bar{y}.\bar{z}$ (x will vanish)

Group 2: $x.\bar{y}.\bar{z}$ and $x.y.\bar{z}$ so simplified expression is $x.\bar{z}$ (y will vanish)

Step 4: Write the final simplified form as a sum of products, the ungrouped term will be added as it is

$$f(x, y, z) = \bar{y}.\bar{z} + x.\bar{z} + \bar{x}.y.z$$

Example 2: Simplify the Boolean function

$$f(x, y) = \bar{x}.y.z + \bar{x}.y.\bar{z} + x.y.z + x.y.\bar{z} + x.\bar{y}.\bar{z}$$

Solution:

Step 1: Represent the function in the form of a k-map. This is shown below

$x\backslash y.z$	$\bar{y}.\bar{z}$	$\bar{y}.z$	$y.z$	$y.\bar{z}$
$\bar{x}$	0	0	1	1
x	1	0	1	1

Step 2: Mark any groups of two or four adjacent 1 as shown below

$x\backslash y.z$	$\bar{y}.\bar{z}$	$\bar{y}.z$	$y.z$	$y.\bar{z}$
$\bar{x}$	0	0	1	1
x	1	0	1	1

The groups are

Group 1: $\bar{x}.y.z$, $x.y.z$, $\bar{x}.y.\bar{z}$, $\bar{x}.y.\bar{z}$

Group 2: $x.\bar{y}.\bar{z}$ $x.y.\bar{z}$

It is extremely important to note that the squares on the left edge are taken to be adjacent to the squares on the right edge. These form the group 2 and have been marked by using rectangular shape

Step 3: Write simplified expression for each group.

The simplified form of group 1 is y because both x, $\overline{x}$, and z, $\overline{z}$ are changing in the group.

Also the simplified form of group 2 is $x \cdot \overline{z}$ because both y and $\overline{y}$ are changing in the group.

Step 4: Write the final simplified form as a sum of products

$$f(x, y) = y + x \cdot \overline{z}$$

Example 2: Simplify the Boolean function

$$f(x, y) = x.y.\overline{z} + \overline{x}.\overline{y}.\overline{z} + x.\overline{y}.\overline{z} + \overline{x}.y.z + \overline{x}.\overline{y}.z + \overline{x}.y.\overline{z} + x.y.z$$

Solution:

Step 1: Represent the function in the form of a k-map. This is shown below

$x \backslash y.z$	$\overline{y}.\overline{z}$	$\overline{y}.z$	$y.z$	$y.\overline{z}$
$\overline{x}$	1	1	1	1
x	1	0	1	1

Step 2: Mark any groups of two or four adjacent 1 as shown below

$x \backslash y.z$	$\overline{y}.\overline{z}$	$\overline{y}.z$	$y.z$	$y.\overline{z}$
$\overline{x}$	1	1	1	1
x	1	0	1	1

so there are three groups

Group 1: $\overline{x}.\overline{y}.\overline{z}$ $\quad \overline{x}.\overline{y}.z$ $\quad \overline{x}.y.z$ $\quad \overline{x}.y.\overline{z}$ (Top Row)

Group 2: $\overline{x}.y.z$ $\quad \overline{x}.y.\overline{z}$ $\quad x.y.z$ $\quad x.y.\overline{z}$ (Last two columns)

Groupt 3: $\overline{x}.\overline{y}.\overline{z}$ $\quad x.\overline{y}.\overline{z}$ $\quad \overline{x}.y.\overline{z}$ $\quad x.y.\overline{z}$ (First and last column)

Once again note that the squares on the left edge are taken to be adjacent to the squares on the right edge. These form the group 2 and have been marked by using rectangular shape. Also note that a minter can be used in more than one group.

Step 3: Write simplified expression for each group.

Group1: becomes $\overline{x}$
Group2: becomes y
Group3: becomes $\overline{z}$

Step 4: Write the final simplified form as a sum of products

$$f(x, y, z) = \overline{x} + y + \overline{z}$$

You can also notice that a group of two 1's eliminates one literal, a group of four 1's eliminates two literals and a group of eight 1's eliminates three literals. So if all the squares have 1's then all literals are eliminated and function becomes constant i.e., 1

Advantages and Disadvantages of k-map method

Some advantages of this method of simplification are given below.

- This method is very easy to follow.
- This is a systematic process. It always leads to a single minimal solution.

A disadvantage of this system is that it is not scalable. This means that this system works very well for less variables but becomes complex for higher number of variables

Exercise 6

1. State and prove the **De Morgan's laws** for the Boolean algebra.
2. If x and y are Boolean variables then prove the following identities by using truth table.
   - **a.** $\overline{x} + \overline{y}$
   - **b.** $x + x.y = x + y$
   - **c.** $x.(x + y) = x$
   - **d.** $x + 1 = 1$
   - **e.** $x.0 = 0$
3. Make truth table of the following functions:
   - **a.** $f(x, y) = x.y + \overline{x}.y$
   - **b.** $x.\overline{y} + \overline{x}.y$
4. Calculate the value of the following Boolean functions at the given values of x, y and z.
   - **a.** $\overline{x}.y + \overline{x}.\overline{z} + x.\overline{y}$ for $x = 0$, $y = 1$ and $z = 0$
   - **b.** $(\overline{x} + y).x + (\overline{y} + z)$ for $x = 0$, $y = 1$ and $z = 1$
5. Prove the following results and apply the principle of duality to obtain the dual of these results.
   - **a.** $x + \overline{x} = x$
   - **b.** $x + 0 = x$
   - **c.** $\overline{x} + x.y = \overline{x} + y$
   - **d.** $\overline{x}.(y + z) = (\overline{x}.y) + (\overline{x}.z)$
6. Explain the following Logic gates and show their function by using a truth table.
   - **a.** AND
   - **b.** OR
   - **c.** NOT
7. Represent the following Boolean expressions as a combination of logic gates.
   - **a.** $x.\overline{y} + \overline{x}.y$
   - **b.** $x.\overline{y} + \overline{x}.y$
   - **c.** $\overline{x} + \overline{x}.y$
8. Simplify the following Boolean functions using K-maps :
   - **a.** $f(x, y) = x + \overline{x}.y$
   - **b.** $f(x, y, z) = \overline{x}.y.z + x.\overline{y} + \overline{x}.\overline{y}.z$
   - **c.** $f(x, y, z) = x.z + \overline{x}.z.y$
9. Fill in the blanks.
   - (i) Commutative laws states that $a + b$ is equal to__________.
   - (ii) By distributive law we know that $ab + ac$ is equal to __________.
   - (iii) A+ 0 is equal to__________.
   - (iv) 0 is called the __________.
   - (v) Boolean Algebra operates on __________.

(vi) In Boolean algebra the identity element with respect to dot (.) is ______

(vii) $x + x$ is equal to ______

(viii) ______ is a very efficient way of solving Boolean functions.

(ix) $\overline{x \cdot y}$ is equal to ______

(x) In Boolean algebra standard product is called ______.

10. Match the following

$(a+b)$	Sum of products
Minterms	$x.0=0$
Maxterms	Product of Sums
$x+1=1$	$a.b$

11. Choose the correct answer.

(i) K-Map is used to
 a. Evaluate a Boolean expression **b.** Simplify a Boolean expression
 c. Both a and b **d.** Non of above

(ii) Demorgan's Law states that
 a. $a(b+c) = a.b + a.c$ **b.** $a + (b+c) = (a+b) + c$ **c.** $\overline{a+b} = \overline{a} \cdot \overline{b}$ **d.** none of above

(iii) A Boolean function with four variables will have
 a. 8 maxterms **b.** 16 maxterms **c.** 24 maxterms **d.** 32 maxterms

(iv) The idempotent law states that for two variables x and y
 a. $x + x \cdot y = x + y$ and $x \cdot (x + y) = x$ **b.** $\overline{\overline{x}} = x$ **c.** $x.x = x$ and $x + x = x$
 d. none of the above

(v) The absorption law states that for two variables x and y
 a. $x.x = x$ and $y.y = y$ **b.** $x. \ \overline{y} = \overline{y} .x$ **c.** $x + x \cdot y = x + y$ and $x \cdot (x + y) = x$
 d. none of the above

12. Mark the following as True/False.

(i) Idempotent law states that $x+1 = 1$.

(ii) K-map is used to simplify a Boolean expression. (iii) $x + y + z$ is a minterm.

(iv) k-map may or may not lead to a single minimal solution.

(v) A Boolean function cannot involve more than two variables.

(vi) The principle of duality states that . and + are interchangeable.

(vii) As the number of variables in a Boolean function increase, the k-map becomes more complex.

(viii) A Boolean function involving 5 variables will have 31 minterms.

(ix) To simplify a k-map of groups of two, four, six, or eight 1s can be marked.

(x) Involution principle states that $y + \overline{y} = 1$.

Answers

Q.9

(i) $b+a$	(ii) $a.(b+c)$	(iii) A	(iv) additive identity	(v) Binary numbers
(vi) 1	(vii) x	(viii) K-map	(ix) $\overline{x} + \overline{y}$	(x) minterm

Q.11

(i) c	(ii) c	(iii) b	(iv) c	(v) c

Q.12

(i) F	(ii) T	(iii) F	(iv) F	(v) F
(vi) T	(vii) T	(viii) F	(ix) F	(x) F

Chapter 7

Computer Software

7.1 Introduction

In previous chapters, we discussed about different parts and configurations of the computer. It has been mentioned that programs or instructions are given to the computer to do a specific task. So it is necessary to provide sequence of instructions so that your work can be done. **Software** is a single/collection of programs that performs a particular task. A computer is useless without software and vice versa. In this chapter, you will learn about different types of commonly used software and their functions. There are two types of software on the basis of functionality: *System Software and Application Software.*

The **Application Software** is a program created to perform a specific task for a user. For example, to create a document, a word processing software is used such as Microsoft Word. Whereas, to create spreadsheets, we can use Excel or Lotus 123, etc.

7.2 System Software

The software used to control, monitor, or facilitate the use of the computer is called **System Software**.

System Software provides the interface to the computer and controls basic operations like saving data on the disk, making computer to work for our use, printing a document, etc. The system software includes the operating system, the language translators, the linkers, loaders and other utility programs, etc.

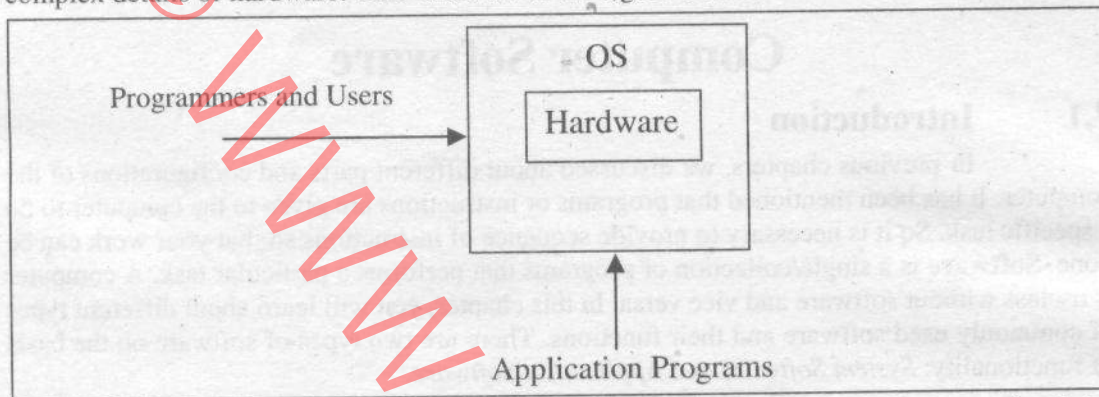
Operating System

In order to use computer, to solve some problem, a user has to write instructions (program). But other than writing instructions for solving the problem every programmer will also have to write instructions for the following tasks:

- Read data from the input devices.
- Show results on the output devices.
- Perform memory management i.e, load programs and data into memory and manage memory.
- Organize data on the storage devices.

These tasks are very complex and only expert programmers can write these instructions. To make the use of computer easy, these programs are provided in the form of a software, known as Operating System (OS).

An operating system is a set of programs running on a computer system, and provide an environment in which other programs can be executed and the computer system can be used effectively.

The operating system provides the most common functions needed by any user. In this way the operating system wraps around the hardware and saves its users from the complex details of hardware. This is shown in the figure below.

From this figure it is obvious that the OS will not only provide programs for doing different tasks but will also provide an interface to its users (i.e. programs, programmers etc).

Types of Operating System

Depending upon the working, the Operating Systems are divided into the following main types:

- Batch Processing Operating Systems.
- Real-Time Operating Systems.
- Single User Operating Systems.
- Multi-User Time Sharing Operating Systems.
- Network Operating Systems

7.3 Functions of an Operating Systems

Following are the main functions performed by the most Operating Systems:

- Manage Hardware Resources
- Load and Execute programs
- Memory Management
- Secondary Memory Store Management
- Providing interface to the users

There are two commonly used user interfaces:

Command Line Interface: In such interfaces the users communicate with the Operating System by typing commands using a keyboard. Each command given to the OS activates one of the many programs in the OS. Example of such an interface is the command prompt provided by MS-DOS to its users.

Graphical User Interface (GUI): The GUI interface consists of Window, Menus, Icons and pointers. The user of the system communicates with the Operating System by selecting commands from the menus or by selecting different icons by using a pointing device like mouse. MS-Windows is a well-known example of an Operating System with a GUI interface.

In MS-Windows the user selects commands by using a mouse and a keyboard e.g. Windows XP, Linux etc.

7.4 Language Translators

This is another very important category of system software and has played a very important role in the development of general-purpose computers. There are three main types of language translators ie, assemblers, compilers and interpreters.

7.4.1 Assembler

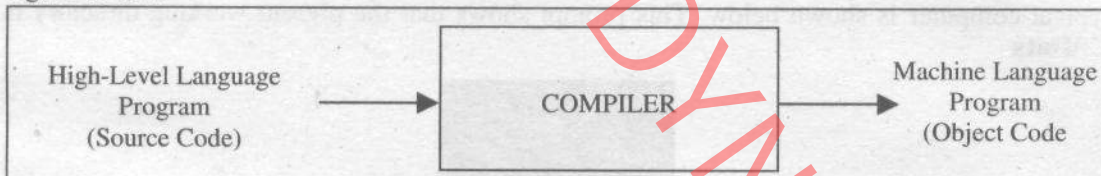
Assembler is a program that converts an assembly language program into machine language.

In the early days of computers, the programs had to be written in machine code instructions using binary codes. So writing programs was a very difficult and time-consuming task. It was also prone to errors and removing these errors was another difficult task.

Assembly language was developed to make the job easier. Assembly language allowed the use of symbolic codes for machine instructions; these codes are called mnemonics (pronounced as Ne-Monics). Obviously, writing programs in Assembly language is much easier than writing instructions in binary. An Assembly language provides one line of code for each machine instruction.

7.4.2 Compiler

A **compiler** is a program that translates a source program (written in some high-level programming language) into machine code as a whole.

The generated machine code can be later executed. A compiler first reads the whole program before executing it. The errors in the code are pointed out and then the machine language code is generated. This output is known as **object program**. The object program executes very fast as it is directly understandable by the computer. Today most of the languages use compilers. Once a program has been translated into machine code it can be loaded into the machine and can be executed. This process of translation is shown in the figure below.

As shown in the above figure, the high level program is called the **source code** and the translated program is called the **object code**

7.4.3 Interpreter

An **interpreter** looks at each line of the source program, decides what that line means, checks it for possible errors and then executes that line.

If one of the lines is executed repeatedly, it must be scanned and analyzed each time, greatly slowing down the solution of the problem at hand. Thus interpreter executes code line by line. If an error is encountered, the execution is stopped at that line and user is notified of the line where the error occurred. After the error has been removed, the user has to run the whole program from the beginning. This whole process slows down the performance of the interpreter as compared to the compiler. Most of the languages used for writing short scripts of code use interpreters.

7.5 Disk Operating System (DOS)

It is a single user operating system and has been very popular on microcomputers up to mid 1990s. DOS was designed by IBM (International Buisness Machines). DOS resides on

disk and controls the overall functioning of the computer. It performs the following major tasks:

- Control input and output devices
- Execute user programs
- Manage system resources
- Provide user interface
- Memory management

DOS does not provide networking features. To connect a computer running on DOS, some third party networking software should be installed.

7.5.1 DOS Files

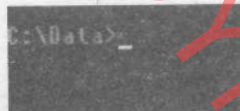
There are three important files in DOS which it uses for different purposes. These are **Batch Files, Command Files** and **Executable Files.** These files are identified by their extensions which are .bat, .com, and .exe respectively.

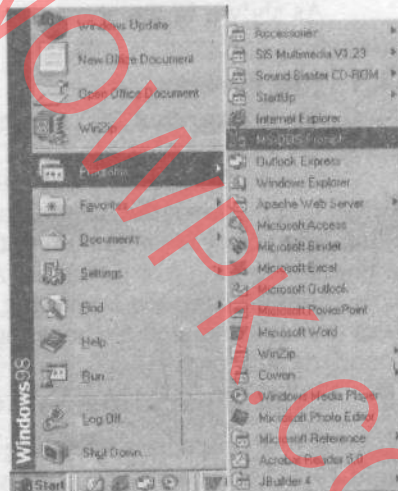
In **Batch Files**, one or more commands are grouped together. The name of batch file acts as a command for the DOS. When the name of file is given to DOS on command prompt, the commands written in the batch file are executed sequentially without involving user to type every command on command prompt.

Executable Files are in executable form i.e. these are ready to run on the computer. These files contain instructions in machine language whereas **Command Files** contain DOS commands.

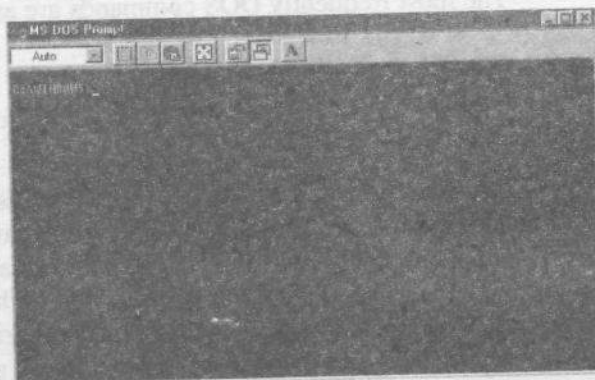
7.5.2 User Interface of DOS

DOS provides a command driven interface to the users. In a command driven interface the users can use the keyboard to type the command on the command prompt. Mostly the DOS prompt shows the present working directory. DOS interprets all commands with reference to the present working directory until explicitly stated. The DOS prompt on a typical computer is shown below. This prompt shows that the present working directory is **C:\Data**

```
C:\Data>_
```

In Desktop computers Microsoft Windows is the most commonly used operating system and DOS is no more in common use. Windows also provide a command prompt similar to the DOS prompt and provide all commands used in DOS. Actually **Windows 9.X** operating system's work on top of DOS **Windows NT** and **Windows 2000** also provides its users with a command prompt and commands used in DOS. So learning the DOS commands can be very useful.

If you have DOS installed as the main Operating System then the DOS prompt is automatically shown when you switch on your computer. If you have **Windows 98** installed on your machine, then you can go to the DOS prompt as shown in the above figure and explained below.

To activate the DOS prompt, first click at the start button shown on the desktop of your computer. Then choose the Programs menu by using the mouse or the keyboard. From this menu choose MS DOS prompt option. You will see the following Screen.

To expand the prompt to full screen, press the **ALT + Enter** keys. Note that you can restore to the original screen by pressing these keys again. It is also important to note that you can close this screen by entering the **Exit** command on the prompt.

If you have **Windows NT** or **Windows 2000** installed on your computer, then you can use a similar procedure to locate the command prompt.

7.6 DOS Commands

There are two types of DOS command:

- Internal Commands
- External Commands.

DOS **internal commands** are stored in the COMMAND.COM file. These are loaded automatically into the memory during booting process. These include the commonly used commands. Internal commands are part of Command.com thus you see their names in directory listing. These commands remain in memory during the execution of DOS. Some of the internal DOS commands are: CLS, DIR, DEL, DATE, TIME, EXIT etc.

DOS **external commands** are those, which need special files for their execution. The DOS commands which are not frequently used are given as external commands. The three types of files that can run as an external commands are COM, EXE and BAT files. Some of the external DOS commands are: CHKDSK, DELTREE, FORMAT, XCOP etc.

How does DOS Organize Data?

DOS divides the secondary storage of the computer into logical areas called drives and each drive is recognized by a letter symbol followed by a colon. For example, the one floppy disk on your computer may be called **A:** and second floppy disk may be called **B:** drive. The hard disk may be assigned the letter C and is called **C:** drive and the compact disk may be called the **D:** drive. It is very common that the user may have a very large hard disk and divides it into logical parts called partitions. Each partition is also assigned a drive letter. For example, if you have a 20 GB hard disk, then you may use the **FDISK** utility to divide it into four partitions of equal size. In this case your floppy disk will be called the A: drive and four partitions of your hard disk may be called **C:, D:, E:,** and **F:** drives.

The data on each drive is organized in the form of directories and files. For example, you may have four subdirectories named **DOS, Command, Windows** and **Data** on drive **C:** of your computer. You may create more directories on this drive or within the existing subdirectories. You can save data in the form of files on drive **C:** or in any of the directory on C:

At any given time DOS marks on of the directory as the working area and executes all commands with respect to that directory until specified explicitly. Such a directory is called the **present working directory.**

The most frequently DOS commands are as follows:

7.6.1 CD (Change Directory)

CD the most frequently used DOS command. This command is used to change the present working directory.

Suppose that you have a game PACEM installed on my computer on drive C: and in the directory GAMES. Then all the files of this game are in C:\GAMES\PACEM\ directory. This is called the path of these files. The path actually tells the logical location of the files on the disk. The PACEM.EXE file is also located in this directory and you need to load this file into memory and execute it to play the game. Before you start playing this game it is always a good idea to make C:\GAMES\PACEM\ directory as your present working directory as some executable files use the present working directory for different purposes.

Assume that the present working directory is C:\DOS. To change the present working directory from C:\DOS> to C:\Games\Pacem you can enter the following sequence of commands on the prompt.

C:\>CD ..
C:\>CD GAMES
C:\>CD PACEM

Or we can type the single command CD C:\Games\Pacem.

Note: In this command we have specified the full path of the directory.

What happens if a directory or path you have specified within your CD command does not exist? The answer is bad command or invalid directory.

Following examples show the use of this command:

Example 1: To change the directory to the parent directory you can use the following command.

C:\GAMES\PARA

This means that if your present working directory is **C:\GAMES\PARA**, then you can use **CD..** to change the present working directory to **C:\GAMES>**

Now suppose that the present working directory is **C:\GAMES\PARA** and you enter the following command:

CD D:\TEST , your current directory will become **Test** in **D:** drive.

Example 2: To change the present working directory from C:\GAMES\PACEM to D:\TEMP you can enter the following sequence of commands at the prompt.

D:
CD TEMP

Note that to change the present drive from C: to D: you need to enter the drive name followed by colon at the command prompt.

Example 3: To change the present working directory to the root-drive you can use the following command:

**CD **

So if the present working directory is C:\GAMES\PACEM, then to change it to C:\ you can use the above command i.e. CD \

Example 4: Suppose the present working directory is C:\GAMES\ and this directory contains a subdirectory PACEM, then to change the present working directory to C:\GAMES\PACEM you can enter the following command at the prompt.

CD PACEM

Note that in this case you do not need to specify the full path of the directory.

Example 5: To change the present working directory from C:\GAMES to C:\GAMES\PACEM\TEMP\ you can enter the following command.

CD PACEM\TEMP

The general syntax of Change Dirctory (CD) command is **CD[DRIVE:]PATH**

If we do not specify the drive name before the path, then path is taken to be from the present working directory.

Now that you know how to move in different directories, we can learn other important commands.

7.6.2 MD/(MKDIR)

This command is used to create subdirectories e.g. if the present working directory on your computer is C:\GAMES, then you can enter the following command to create a directory named NEWGAME within C:\GAMES directory.

MD NEWGAME

Now if you want to create another directory within NEWGAME subdirectory, then there are two possibilities.

You can use CD command to change the present working directory to NEWGAME and then use MD to create a subdirectory or you can create the directory by giving the command

MD NEWGAME\TEMP

In this case we do not need to change the present working directory. You can also give following command to make a new directory named **TEST** within **D:\TEST\TEMP**

MD D:\Test\Temp\Test

7.6.3 RD /RMDIR (Remove Directory)

If we have an empty directory and wand to delete it RD can be used the most general syntax of writing this command is :

RD [DRIVE][PATH] or RMDIR [DRIVE][PATH]

Example: To delete the directory TEMP from the directory C:\DATA you can enter the following command on the prompt

RD C:\DATA\TEMP
or
RD TEMP if the present working directory is C:\DATA

RD removes empty directories only. If a directory contains a file or a subdirectory, then RD does not delete this directory.

7.6.4 DIR

This is one of the most frequently used DOS command. It is used to display a list of the files and subdirectories that are in the directory. For example, to see the subdirectories and files in the present directory we can enter the following command on the command prompt.

DIR

The output of this command is shown in the adjoining figure.

Note that the command was executed from the directory **C:\Data** and it displays the following important information about this directory.

- The disk's volume label and serial number and the directory information.
- One directory or filename per line, including the extension.
- The file size in bytes.
- The date and time the file was last modified.
- The total number of files listed and their cumulative size.
- The total number of subdirectories listed.
- The free space (in bytes) remaining on the disk.

You can view the directory entries of any directory by using this command. For example, to view the contents of the directory **C:\DATA\PACEM** on your computer you can use the command

DIR C:\DATA\PACEM

So the general syntax of the **DIR** command is

DIR [DRIVE:][PATH]

It is very important to note that if the drive name is not specified, then the path is taken from the present working directory.

If the present working directory is **C:\DATA** and you enter the command **DIR PAPERS\FINAL** at the prompt, then the contents of the directory **C:\DATA\PAPERS\FINAL** will be shown and contents of the directory **C:\PAPERS\FINAL** will not be shown.

You can also use the **DIR** command to see the existence and properties of a file in a directory. For example, to see the properties of the file **ABC.TXT** in the directory **C:\Data\Backup** you can use the following command at the command prompt.

DIR C:\Data\Backup\ABC.TXT

If the file and path of the files both exits, then the **DIR** command will display its properties on the screen.

Using Wildcards with DIR

Very often we want to search for a certain file based on some known property of the file name. For example, we may want to list all executable (. **EXE**) files in a directory or we may want to see only those files which have name starting with **car.** To list such files we can use wildcard characters * and ? are used as wildcard characters and have the following meanings:

- * is used to denote any number of characters.
- ? is used to denote exactly one missing character.

Example 1: Write a DIR command to show all files having .EXE as their extension in the directory DATA\GAMES\PACEM on the C: drive of your computer

Solution: As we are looking for all files having extension **.EXE** and the file can have any name so we can use the following command,

DIR C:\DATA\ GAMES\PACEM *.EXE

In this command the drive-name is **C:** the directory is **\DATA\ GAMES\PACEM ** and the file name is * and extension is **.EXE.** As mentioned earlier the * means any number of characters so all the files having extension **.EXE** will be shown.

Example 2: To view all files in the present working directory which have name starting with **CE** you can use the following command.

DIR CE*.DAT

This command tells the DOS to show all files having names starting with the characters **CE** followed by any characters and then a **.** and then the extension **DAT**

Example 3: To view all files in the present working directory which have name containing the characters **"CE"** and having the character **A** in its extension you can use the following command.

DIR *CE*.*A*

This command tells DOS to show all file names starting with any character followed by the characters **CE** followed by any characters and then a dot (.) followed by any characters followed by the character **A** followed by any character

Example 4: To view all files in the present working directory which have name ending with character **X** you can use the following command.

DIR *X.*

Example 5: To view all files, in the present working directory, which have a four-character name you can use the following DOS command.

DIR ????.*

Note that one ? denotes one arbitrary character so ???? means four characters.

Example 6: To view all files, in the present working directory, which have a **X** as third character of the file name the following command can be used.

DIR ??X*.*

Example 7: To view all files, in the present working directory, which have a **X** as third last character of the file name you can use the following command.

DIR *X??.*

We can similarly use these wildcard characters to write many useful commands.

From the above discussion, it is obvious that the most general syntax of the **DIR** command is

DIR [DRIVE:][PATH] [FILENAME]

where

[DRIVE:][PATH] specifies the drive and directory for which we want to see a listing and

[FILENAME] specifies a particular file or group of files.

Using Switches with DIR

Often the directories contain many files and the **DIR** command cannot show the directory information on one screen. In such cases it is very hard to view the directory information as the directory information scrolls up and we do not get enough time to view the files. In such situations we can enter the following version of DIR command on the prompt.

DIR / P

The **/P** is called a switch and it makes the **DIR** command to display it's output one screen at a time and display the message **Press any key to continue**. So the user can view the information one full screen at a time by using the keyboard.

The switches used with the **DIR** command are given in the following table:

Switch	Purpose
/P	Displays one screen of directory list at a time.
/W	Displays the listing in wide format
/A[[:] [attributes]	Displays only the names of those directories and files with the attributes you specify. If you do not use this switch, DIR displays the names of all files except hidden and system files. If we use this switch without specifying any attributes, DIR displays the names of all files, including hidden and system files. We can specify the following Attributes of the file A. Files having Archive variable set are shown H Only hidden files are shown R Read only files are shown by using this option S System files are shown using this option
/O[[:] [sortorder]	Controls the order in which DIR sorts and displays directory names and filenames. If this switch is not used DIR displays the names in the order in which they occur in the directory. We can use this switch without specifying a sort order to sort the output in alphabetic order We can also specify other sort orders of our choice. Following is the list of choices for sorting the output of IR command N In alphabetic order by name

Switch	Purpose
	-N In reverse alphabetic order by name (Z through A) E In alphabetic order by extension -E In reverse alphabetic order by extension (Z through A) D By date and time, earliest first -D By date and time, latest first S By size, smallest first -S By size, largest first G With directories grouped before files -G With directories grouped after files
/S	Lists every occurrence of a file or directory in the specified directory and all of its subdirectories. This switch can be used to search a file.
/B	Lists each directory name or filename, one per line. This switch displays no heading information and no summary.
/L	Displays unsorted directory names and filenames in lowercase.

Example 1: To list all hidden files in the directory **C:\TEMP** use the following **DIR** command .

DIR C:\TEMP / AH

Example 2: To list all the files, which are read-only and hidden following **DIR** command can be used.

DIR / ARH

Example 3: To view all files and directories including the hidden and system files and directories from the present working directory, enter the following command on the prompt.

DIR /A

Example 4: Suppose you want **DIR** to display one directory listing after another, until it has displayed the listing for every directory on the disk in the C drive. Suppose that you also want DIR to alphabetize each directory listing and display it in wide format, and pause after each screen. For such a display you can use the following command.

DIR C:\ /S/W/O/P

DIR lists the name of the root directory, the names of the subdirectories of the root directory, and the names of the files in the root directory (including extensions). Then DIR lists the subdirectory names and filenames in each subdirectory.

Example 5: To list only the subdirectories in the present working directory. For this we can enter the following command on the prompt.

DIR /AD

Example 6: To list only the files in the present working directory and not the subdirectories you can use the following command.

DIR /A-D

Example 7: Suppose you want to save output of the DIR command in a file named **DIR.DOC.** You can do it by using the following command:

DIR > DIR.DOC

If **DIR.DOC** does not exist, MS-DOS creates it. This technique can be used with many DOS commands.

Example 8: To display a list of all the filenames with the **.TXT** extension in all directories on drive C we can type the following command:

DIR C:*.TXT /W/O/S/P

DIR displays in wide format, an alphabetized list of the matching filenames in each directory and pauses, each time the screen fills, until you press a key to continue.

Similarly we can use the DIR command and it's switches in many useful ways.

7.6.5 ATTRIB

Before we learn this important command it is important to know the meaning of attribute of a file. In DOS a file is one or more of the following type.

Read only file : The user can read such a file but cannot modify the file

Hidden File : The usual DOS commands are not applied on these files. So if a user gives a command to show files in a directory then the hidden files are not shown.

Archived File : The files have been archived. This means a backup of these files have been taken.

System Files : These files are used by the operating system to perform different functions. Device drivers are example of such files.

DOS uses four variables {A, R, S, and H} to assign four attributes to each file. If a file has been archived {a historical backup has been taken} then the variable **A** is not set. This indicates that the file does not need to be archived or it has not been modified since last archive operation. If the file is read only then the variable **R** is set. If the file is a system, then the variable **S** is set and if the file is hidden, then the variable **H** is set.

The **ATTRIB** command is used to view/change the attributes of files. For example, if the present working directory is **C:\DATA,** then following command can be used to display attributes of all files in this directory

ATTRIB or **ATTRIB *.***

The results of this command are shown in the figure below.

Output of ATTRIB command

Figure

In the above figure, the file OS.DOC is read only, the file QUESTION.DOC is Archive, hidden and read only and the file **Report. DOC** is read only.

To display attributes of all files in the directory **C:\Data\Pacem** you can use the following syntax of the **ATTRIB** command.

ATTRIB C:\Data\Pacem*.*

The general syntax of this command is **ATTRIB [Drive:][Path][FileName]**

You can also use this command to view the attributes of a single file. For example, to view the attributes of a file named **pacem.exe** located in the directory **C:\Data\Pacem** you can use the following command.

ATTRIB C:\Data\Pacem\pacem.exe

You can also use **ATTRIB** to change the attribute of a file. For example, to make a file **REPORT.DOC,** in the present working directory, read-only you can use the following command.

ATTRIB +R REPORT.DOC

Similarly to reset the read only attribute you can use the following command.

ATTRIB -R REPORT.DOC

Here +R and –R are called the switches of the command. The following table gives the switches used with **ATTRIB** command.

Switch	Purpose
+R	Sets the Read-Only file attribute.
-R	Clears the Read-Only file attribute.
+A	Sets the Archive file attribute.
-A	Clears the Archive file attribute.
+S	Sets the file as a System file.
-S	Clears the System file attribute.
+H	Sets the file as a Hidden file.
-H	Clears the Hidden file attribute.

You can also set or reset the attributes of all the files using a single command given below.

ATTRIB –R or **ATTRIB –R C:\DATA**

If a file is hidden or system, then **ATTRIB** will not change its attribute as it is necessary to change these attributes before changing any other attribute.

Another switch used with this command is **/S,** by using this switch we can processes files in the current directory and all of its subdirectories.

We can use these wildcards (as described in the DIR command) in the ATTRIB command. For example, the following command will set the read only attribute of all files having name starting with C .

ATTRIB +R C*.*

We can also set the attributes of directories but cannot use wildcard characters for naming directories.

7.6.6 DEL (Erase)

It is another useful but dangerous command provided by DOS. It is used to delete the files from the disk. Following examples show the use of this command.

Example 1: To delete the CAT.TMP file from the TEST directory on drive C we can use either of the following commands:

DEL C:\TEST\CAT.TMP
ERASE C:\TEST\CAT.TMP

The general syntax of this command is

DEL [DRIVE:][PATH]FILENAME [/P]
ERASE [DRIVE:][PATH]FILENAME [/P]

where

[DRIVE:][PATH] specifies the drive and directory for which we want to see a listing and
[FILENAME] specifies a particular file or group of files.

When **DRIVE** is not specified, then the **PATH** is taken relative to the current working directory. When the **PATH** is not specified, then the file is searched and deleted from the present working directory.
It is important to note that wildcard characters, as described in the **DIR** command, can also be used to specify the file names but it is a very dangerous practice sometimes, so these characters should be used with care.
This command has only one switch /P. If we use the /P switch, DEL displays the name of a file and prompts with a message in the following format:
FILENAME, DELETE (Y/N)?

We can press **Y** to confirm the deletion and **N** to cancel the deletion. IF wildcard characters were used to specify more than one file the message is shown for the next **FILENAME.** We can also press the CRTL+C keys to stop **DEL** command.

Following examples show the use of this command.

Example 2: To delete a file named **ABC.DAT** from the present working directory you can use one of the following commands.
DEL ABC.DAT or **ERASE ABC.DAT**

We can delete more than one file by giving a single **DEL** command as shown in the following example.

Example 3: To delete all the files from the directory named **TEST** on drive **C**, you can use either of the following commands:
DEL C:\TEST or **DEL C:\TEST*.***

When this form of DEL is used DEL displays the following prompt:
All files in directory will be deleted! Are you sure (Y/N)?

We can press **Y** and then **ENTER** to delete all files in the current directory, or press **N** and then **ENTER** to cancel the deletion. The following simple command can delete all files from the present working directory

DEL *.*

It is a very common error that users unintentionally delete all files so this command must be used with care. It is also important to note that this command can not be used to delete directories and if we try to delete a directory using this command all files within that directory are deleted but the directory itself is not deleted.

7.6.7 COPY

This is another very useful command provided by DOS. It is used to make copies of existing files. This command can also be used to combine multiple files into a target file. Following examples show the use of this command:

Example 1: The following command copies a file **MEMO.DOC** as **LETTER.DOC**. Both the files are in the current working directory

COPY MEMO.DOC LETTER.DOC

Example 2: To copy the **NOTE.TXT** file from the present working directory to the directory **C:\DATA** you can use the following command

COPY NOTE.TXT C:\DATA\NOTES.TXT

If the destination directory already contains a file named **NOTE.TXT** then DOS shows the following message.

OVERWRITE C:\DATA\NOTE.TXT: (YES/NO/ALL)

You can press **Y** and **Enter** to overwrite, **N** and **Enter** to cancel and **A** and **Enter** to overwrite all files. The last option is useful when more than one file are being copied.

You can use wildcard characters to specify more than one file to be copied. This is shown in the following example.

Example 3: To copy all files from **C:\DATA\PACEM** into the present working directory you can use the following command:

COPY C:\DATA\PACEM*.*

To copy data into the present working directory you do not need to specify the destination directory in the COPY command.

Example 4: To copy all files from the present working directory into the directory **C:\DATA \BACKUP** you can use the following command:

COPY *.* C:\DATA \BACKUP

If the destination directory already contains some files with the same name, then DOS shows the following message the overwrite message :

OVERWRITE C:\DATA\FileName :(YES/NO/ALL)

You can press **Y** and **Enter** to overwrite, **N** and **Enter** to cancel and **A** and **Enter** to overwrite all files.

Example 5: You can combine several files into one file by using the following **COPY** command.

COPY ONE.TXT + TWO.TXT + THREE.TXT FOUR.NEW

This command will create a new file named **FOUR.NEW** by combining the existing files **ONE.TXT**, **TWO.TXT** and **THREE.TXT** into a new file **FOUR.NEW** we can give the following command

You can also use wildcard characters to specify multiple files to be combined. For example to combine all TXT files in the present working directory into a new file named **UPDATE.TXT** you can use the following command:

COPY *.TXT UPDATE.TXT

It is important to note that if we do not specify a destination file, MS-DOS creates a copy with the same name in the present working directory. If the file to be copied is also in the present working directory then the COPY command stops and MS-DOS displays the following error message:

File cannot be copied onto itself
0 File(s) copied

7.6.8 DATE

This command is used to display the date and it prompts you to change the date if necessary.

The syntax of the Date command is **DATE [mm-dd-yy]**

This command sets the date we specify. Values for day, month, and year must be separated by periods (.), hyphens (-), or slash marks (/). It is important to note the following limitations e.g.

mm	1	through	12			
dd	1	through	31			
yy	80	through	99	or 1980	through	2099

IF you do not specify the date the DOS asks for the date. If the present date of your system is correct you can simply press the **Enter** key without entering the new date.

7.6.9 TIME

To display the current time or to display a prompt, by which you can change the current time enter the following command at the command prompt:

TIME

If you specify the time in an invalid format, MS-DOS displays the following message.

Invalid time
Enter new time:_

To keep the present time you can simply press the enter key

7.6.10 VER

This command is used to display the MS-DOS version number. To view the version number of MS-DOS on your system enter this command at the prompt:

VER

7.6.11 TYPE

This command is used to displays the contents of a text file on the screen. The original file is not modified when we use this command.

The syntax of this command is

TYPE [DRIVE:][PATH]FILENAME

If we display a binary file (e.g. executable files) using **TYPE** command we will see strange characters on the screen.

To display the contents of a file named WORK.TXT we can use the following command:

TYPE WORK.TXT

If this file does not fit on one screen we can use the MORE switch to see the file one screen at a time as follows:

TYPE WORK.TXT / MORE

7.6.12 FORMAT

The FORMAT command creates a new root directory and file allocation table for the disk. It can also check for bad areas on the disk, and it can delete all data on the disk. In order for MS-DOS to be able to use a new disk the disk must be formatted.

Following examples show the use of this command.

Example 1: To format a new floppy disk, in drive A using the default size, type the following command:

FORMAT A:

Example 2: To perform a quick format on a previously formatted disk in drive A, type the following command:

FORMAT A: /Q

Example 3: To format a floppy disk in drive A, completely deleting all data on the disk, type the following command:

FORMAT A: /U

This is commonly known as the unconditional format.

Example 4: To format a floppy disk in drive A and assign to it the volume label "DATA", type the following command:

FORMAT A: /V:DATA

After formatting a floppy disk, FORMAT displays the following message:

Volume label (11 characters, ENTER for none)?

The volume label can be a maximum of 11 characters (including spaces). If you do not want your disk to have a volume label, press ENTER.

When you use the FORMAT command to format a hard disk drive C:\ MS-DOS displays the following message:

WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE C: WILL BE LOST!
Proceed with Format (Y/N)?_

To format the hard disk, press **Y**; if you do not want to format the disk, press **N**.

Note that it is very dangerous command and you may loose important data if you use this command carelessly.

Following table lists few switches commonly used with the **FORMAT** command

Switch	Purpose
/S	Copies the operating system files IO.SYS, MSDOS.SYS, and COMMAND.COM to a newly formatted disk so that it can be used as a system disk.
/Q	Specifies a quick format of a disk. With this switch, FORMAT deletes the file allocation table (FAT) and the root directory of a previously formatted disk, but does not scan the disk for bad areas. This switch is used to format only previously formatted disks that are in good condition.
/V:lbl	Specifies the volume label. A volume label identifies the disk and can be a maximum of 11 characters. If this switch is omitted or used without specifying a volume label, MS-DOS prompts for the volume label after the formatting is completed.
/U	Specifies an unconditional format of a disk. Unconditional formatting destroys all existing data on a disk and prevents from using the UNFORMAT command..

7.6.13 DISKCOPY

This command is used to make copy of the floppy disk and is not used for hard disk. Its syntax is **DISKCOPY Source Destination.**

For example the following command copies the disk in drive **A:** onto a disk in drive **B:**

DISKCOPY A: B:

The DISKCOPY command prompts you to insert the source and destination disks and waits for you to press any key before continuing.

After copying, DISKCOPY displays the following message:

Copy another diskette (Y/N)?

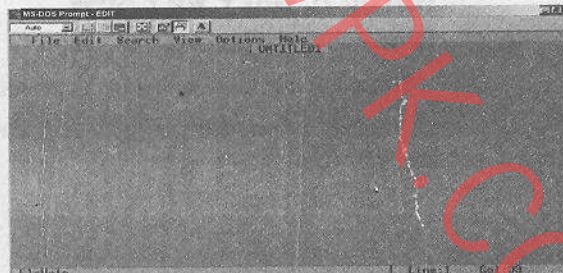
If you press **Y**, DISKCOPY prompts you to insert source and destination disks for the next copy operation. To stop the DISKCOPY process, press **N**.

You can use a single drive as both the source and destination drive. For example the following command copies a disk in drive **A:** onto a disk that will also be inserted in the same drive

DISKCOPY A: A:

DISKCOPY stores an image of the source disk and then copies it after you put the destination disk in the drive. If your computer does not have enough space in the memory or on the hard disk, then you may have to swap the source and destination floppies several times.

7.6.14 EDIT

Edit is the name of an executable file **(EDIT.COM)**. When you enter Edit on the prompt DOS searches for this file and loads it into memory and starts its execution. This program is a text editor that can be used to create and edit ASCII text files. Following screen shows the Edit screen shown by this editor.

MS-DOS Editor allows you to create, edit, save, and print ASCII text files. Following is the syntax for entering this command.

EDIT [DRIVE:][PATH][FILENAME]

If you do not specify a file name then this editor starts a blank file.

7.6.15 SYS

This command is used to transfer the hidden MS-DOS system files (IO.SYS and MSDOS.SYS), the MS-DOS command interpreter (COMMAND.COM), to the disk specified in the command. When you transfer these files on a disk by using this command it can be used to start the system as it contains the DOS commands i.e. COMMAND.COM. The syntax of this command is

SYS [DRIVE1:][PATH] DRIVE2:

where

[DRIVE1:][PATH] specifies the location of the system files. If you do not specify a path, MS-DOS searches the root directory on the current drive for the system files.

DRIVE2: specifies the drive to which you want to copy the system files.

The following example shows the use of this command:

Example: To copy the MS-DOS system files and command interpreter from the disk in the current drive to a disk in drive A, type the following command:

SYS A:

7.6.16 PROMPT

This command can be used to change the appearance of the command prompt. You can customize the command prompt to display any text you want,

Syntax of this command is **PROMPT [TEXT]** where **TEXT** specifies any text you want to show as the prompt.

The following table shows the some interesting character combinations you can use to form your prompt...

Symbol	Text	Symbol	Text	Symbol	Text
$Q	=	$$	$	$T	Current time
$D	Current date	$P	Current drive and path	$V	MS -DOS version number
$N	Current drive	$G	>	$L	<
$B	| (pipe)	$_	Line Feed		

When you use the PROMPT command without specifying a value for text, PROMPT resets the command prompt to the current drive letter followed by a greater-than sign (>).

Following examples show the use of this command:

Example 1: To show my name followed by the > character on your prompt, you can use the following command.

PROMPT Mubasher Baig$g

Example 2: To show the present working directory followed by the > character you can use the following command.

PROMPT pg

Example 3: To display a two-line prompt in which the current time appears on the first line and the current date appears on the second line you can use the following command:

PROMPT TIME IS: T_DATE IS: $D

You can similarly use the symbols given in the above table to display other useful prompts.

7.6.17 DELTREE

We know that the **RD (RMDIR)** command can not delete a directory if the directory is not empty. The DELTREE command can be used to delete the whole directory even if the directory is not empty. This command deletes a directory and all the files and subdirectories that are in it.

The syntax of this command is **DELTREE [/Y] [DRIVE:] PATH**

where **DRIVE: PATH** specifies the name of the directory you want to delete. You can specify more than one directory in a single command.

For example, to delete the directory C:\data\temp you can enter the following command on the prompt.

DELTREE C:\DATA\TEMP

When you use this command DELTREE shows you the following prompt:

Delete directory "C:\DATA\TEMP" and all its subdirectories [YN]

To delete the directory you can enter **Y** and to cancel you can enter **N**.

This command has only one switch **/Y** and when this switch is used DELTREE deletes the directory without first prompting you to confirm the deletion.

You can use wildcards with the DELTREE command, but use them with extreme care. If you specify a wildcard that matches with filenames then files will also be deleted. Following examples show the use of this command.

Example 1: To delete the **TEMP** directory on drive **C**, including all files and subdirectories of the TEMP directory, enter the following at the command at the prompt

DELTREE C:\TEMP

Example 2: To delete the **TEMP** directory on drive **C**, and **TEMP1** directory on drive D, including all files and subdirectories of the TEMP and TEMP1 directory, type the following at the command prompt:

DELTREE C:\TEMP D:\TEMP1

Example 3: To All directories and files having **T** as the first name of their name you can use the following command.

DELTREE T*.*

DELTREE prompts you before deleting any directory and file in this case. To avoid the prompt you can use the following command:

DELTREE /Y T*.*

7.6.18 XCOPY

This command is used to copy **directories**, their **subdirectories**, and files (except hidden and system files). We know that the copy command can not be used for copying the subdirectories so XCOPY command is very useful. Also this command has some useful switches for taking backup of the data. This command is provided in the form of an executable file. **(XCOPY.EXE)**

Following table lists some useful switches used with **XCOPY:**

Switches

Switch	Purpose
/Y	Indicates that you want XCOPY to replace existing file(s) without prompting you for confirmation.
/A	Used to Copy only source files that have their archive file attributes set.
/M	Used to Copy source files that have their archive file attributes set. Unlike the /A switch, the /M switch turns off archive file attributes in the files specified in source
/D: DATE	Used to Copy only those source files modified on or after the specified date.
/S	Used to copy non-empty directories and subdirectories
/E	Used to copy all directories and subdirectories even if the directories or subdirectories are empty.
/V	This switch is used to verify each file as it is written to the destination file to make sure that the destination files are identical to the source files.

Example 1: To copy all the files and subdirectories (including any empty subdirectories) from the present working directory into disk in drive A you can use the following command:

XCOPY *.* A: /S

Example 2: The following command uses the /D: and /V switches

XCOPY C:\Data\Backup: A: /D:01/18/04 /S /V

This command will copy only those files from the **C:\ DATA\BACKUP** directory onto the disk in drive A that were written on or after 01/18/93. After the files are written the XCOPY command compares the files to the source to make sure they are the same.

7.6.19 CHKDSK

This command is used to checks the status of a disk and to fix errors on the disk if any. This command can also be used to see if a certain file has any errors or is stored in contiguous blocks on the disk or not. This command has two switches /F and /V. /V switch is

used to display the files checked by CHKDSK and /F is used to correct error in a file or on the disk. It is important to note that all modern versions of DOS and Windows recommend the use of SCANDISK instead of CHKDSK as SCANDISK can detect and correct a much wider range of disk errors.

To use CHKDSK to check your drive C for errors enter the following command on the prompt.

CHKDSK C:

A sample output of this command when it is executed on a computer is given in the following figure:

```
Volume C     created 05-06-2003 10:04p
Volume Serial Number is 284C-1DD7
Errors found, F parameter not specified
Corrections will not be written to disk

  90 lost allocation units found in 3 chains.
  2,949,120 bytes disk space would be freed

2,146,500,608 bytes total disk space
   74,055,680 bytes in 635 hidden files
   33,128,448 bytes in 1,007 directories
1,291,321,344 bytes in 14,011 user files
  745,046,016 bytes available on disk

     32,768 bytes in each allocation unit
     65,505 total allocation units on disk
     22,736 available allocation units on disk

    655,360 total bytes memory
    615,104 bytes free

     Instead of using CHKDSK, try using SCANDISK.  SCANDISK can reliably detect and fix a much
wider range of disk problems.
```

To find out how much data is stored on drive C and how much space is still free, and to check the disk for errors, type the following command:

CHKDSK C:

7.6.20 PATH

This command is used to specify/view the directories MS-DOS should search for executable files. By default, the search path is the current directory only. This command has the following syntax:

PATH DRIVE: PATH; DRIVE: PATH...

To view the current search path you can use the following command:

PATH

To clear all search-path settings other than the default setting you can use the following command:

PATH;

It is important to note that MS-DOS always searches in the current directory first, before it searches directories in the search path.

You can specify more than one path for MS-DOS to search by separating entries with a semicolon (;)

For example, the following command specifies that MS-DOS is to search three directories to find commands

PATH C:\DOS\; C:\DATA\D:\GAMES; C:\COMMAND

7.6.21 VOL

This command displays the disk volume label and serial number, if they exist. The general syntax of this command is as follows:

VOL [Drive:]

Example:

VOL D:

This will display the volume of D: drive and its serial number

7.6.22 TREE

This command displays the folder structure of a drive or path. The general syntax of this command is:

TREE [Drive:] [Path] [/F] [/A]

/F displays the name of files in each folder

/A use ASCII instead of extended characters

Example:

TREE C:

This will display the folder structure of C: drive.

Exercise

1. Describe (a) System Software and (b) Application Software.
2. What is DOS? How is it different from Windows?
3. How many types of commands are available in DOS? Discuss briefly.
4. What is a language translator? Describe its types briefly.
5. What are switches and wild card? Discuss their uses in DOS with examples.
6. Define operating system. List important functions of operating system.
7. What is the difference between .com and .exe programs?
8. What is a directory, a volume label, and drive name?
9. How do you launch the command processor under Windows XP?
10. Describe the following terms:
    - (i) path
    - (ii) parent directory
    - (iii) subdirectory
11. How do you
    - (i) list all text files in subdirectory b:\reports\
    - (ii) list all files with name accounts under directory a:
12. Suppose you are working in directory C:\testdirectory. How whould you perform the following:
    - (i) create a new directory named **user**
    - (ii) change directory to C: by changing to parent directory twice
    - (iii) delete file named sample3.doc under the testdirectory
    - (iv) Remove the testdirectory after deleting files sample2.txt and sample3.doc.
13. Write DOS commands:
    - (i) To view current date

(i) change date to new date 2004-06-25
(ii) change date back to 2004-06-16 in one statement.

14. Take listing of: (a) all files under testdirectory when you are under C:\ (b) list all files of name sample under the testdirectory and (c) listing of all files of extension .doc under the testdirectory.

15. Write DOS commands to Erase
(i) sample.doc file under C:
(ii) sample4.doc file under testdirectory
(iii) All files under testdirectory.

16. Make testdirectory2 under subdirectory testdirectory when you are under C:\.

17. Explain the following commands:
(i) format
(ii) exit
(iii) find
(iv) pause
print.

18. Change prompt to: (a) current time b) version number 8c) default drive (d) > character and (e) < character.

19. Write down the procedure for writing autoexec.bat file.

20. Explain sort and sys commands

21. Explain type, volume and xcopy commands.

22. Practice all important DOS commands using Windows Command Window.

23. Practice all important internal DOS commands using Windows Command Window.

24. Practice all important external DOS commands using Windows Command Window.

25. Discus in the class room the salient points of MS-DOS.

26. Discus in the class different DOS commands and their switches.

27. Fill in the blanks
(i) ______________ and ______________ convert a high level program into a machine program.
(ii) ______________ provides the command line interface to the users.
(iii) **DIR /p** command is used to ______________
(iv) ______________ command can be used to delete all **.EXE** file from a directory.
(v) ______________ is an external DOS command and ______________ is an internal command
(vi) DOS stands for ______________
(vii) system software are necessary for the ______________ use of computer
(viii) __________ can be used to make a file read-only
(ix) ______________ command can delete folders and all subfolders and files within it.
(x) The ________ switch is used with FORMAT command to conditionaly format a drive.
(xi) FDisk is a(n) ______________ DOS command

28. **Match the following**

DIR	Interpreter
ATTRIB	Compiler
Operating system	View a directory
Line by line translation	Make a file read only
High level language	Memory management

29. **Choose the correct answer**
(i) XCOPY
a. can copy subfolders as well
b. is an external DOS command
c. both a and b
d. none of above

(ii) Windows

a. has a GUI
b. is not an operating system
c. is a compiler
d. all of above

(iii) DOS

a. is an operating system
b. is not an operating system
c. has Graphical user interface
d. none of above

(iv) Interpreter translate

a. the assembly language program line by lin
b. the source program line by line
c. the source program as a whol
d. none of the above

(v) dir ?lass.* command

a. will list all files whose last four characters are **lass** with any extention
b. will list all files starting with any character and with any extension
c. will list all files whose name starts with any character but last four characters are **lass**
d. none of the above

(vi) **rmdir** command

a. makes directory **b.** removes directory **c.** copies file **d.** none of the above

(vii) **prompt** command

a. confirms deletion of file **b.** changes prompt **c.** searches a directory

(vi) **d.** none of the above

(vii) **dir *.***

a. lists all files of extension .doc. **b.** lists all files
c. lists all files of name sample. **d.** None of the above

30. Mark the following as True/False

(i) An Assembler converts a high level language program into machine language
(ii) MD command can be used to delete directories as well.
(iii) **XCOPY /s *.* d:** command copies all the files from the present directory to d:\copy\ directory
(iv) FORMAT a: command will delete all existing data from the a: drive and prepare it for storing data
(v) DOS external commands resides in Command.com
(vi) Diskcopy can not copy files from hard disk
(vii) Batch file contain multiple DOS commands to be executed
(viii) Time command can be used to change the current time
(ix) DELTREE is an internal DOS command
(x) FORMAT command can not format an already formatted disk

Answers

Q.27

(i) Compiler, Interpreter (ii) DOS (iii) List files and directories page wise iv. del *.exe
(v) DELTREE, DIR (vi) Disk Operating System (vii) effective (viii) ATTRIB Command
(ix) DELTREE (x) /u (xi) External

Q.31

(i) c (ii) a (iii) a (iv) b (v) c
(vi) b (vii) B (viii) b

Q.32

(i) F (ii) F (iii) F (iv) T (v) F
(vi) T (vii) T (viii) T (ix) F (x) F

Chapter
8

Introduction to Windows

Microsoft windows is a family of operating systems for personal computers. Windows dominates the personal computer world, running almost on 90% of all personal computers. The remaining 10% are mostly Macintosh computers and LINUX operating system based machines. Windows provides a graphical user interface (GUI), virtual memory management, multitasking, and support for many peripheral devices. This chapter will show you all the basics you need to get started. You'll get an overview of Windows features, and you'll learn how to use Help to get an answer to your questions and find out more about using windows operating systems.

8.1 Main Keywords Associated With Microsoft Windows

- **Disk Drives**
  Drives are devices used to store data. Most computers have atleast two drives: a hard drive C:\ (Which is the main means of storage) and a floppy drive (which stores smaller volumes of data (1.44 Mb) on 3.5" disks - floppy disks). The hard drive is typically designated the C:\ drive and the floppy drive is typically designated the A:\ drive. If you have an additional floppy drive, it is typically designated the B:\ drive. A hard disk can have multiple partitions as well. In that case the first partition will be labeled as C:\ and the rest will be labeled as D:\, E:\ and so on depending upon the number of partitions. You can also have network drives on your computer, depending on authorization and rights of each user. These drives will typically be labeled after the hard drives. e.g. H:\ or G:\

- **Folders (Directory)**
  Folders are used to organize the data stored on your drives. Think of your computer drives as filing cabinets. You can sort your data on drives with folders that store different files. The files that make up a program are stored together in their own set of folders. You can organize the files you create in folders for better accessibility. Folders can be copied and moved from one place to another on your hard disk.

- **File Extensions**
  File Extensions are the ending letters which are placed after a DOT in the file name e.g. *PhoneNumbers.txt*. In this file name, *PhoneNumbers* is the name of the file and *txt* is the file extension. This extension of a file associates it with an application that can be used to view and manipulate it. In this way Windows knows which program to open for a particular file type. For example, a text file has an extension of .txt, so a text file created in Notepad with the name *PhoneNumbers* would look like this *PhoneNumbers.txt*. You do not have to assign a file extension to a file that you create. The program you use will automatically do this for you. All you need to do is give it a filename. Some other common extensions are as follows:
  - .doc = Microsoft Word Document

- .xls = Microsoft Excel Document
- .ppt = Microsoft PowerPoint Presentation
- .mdb = Microsoft Access Database
- .bmp = Windows Bitmap Picture
- .wav = Sound File
- .html or .htm = hypertext document

- **Icon**

  An Icon is a graphic image. Icons help you execute commands quickly. Commands tell the computer what you want the computer to do. These may be shortcuts to applications which are installed on your computer. If you want to execute a command by using an icon, double-click on it. Below are some of the icons and their uses.

My Computer		*My Computer* icon provides access to the different parts on your computer. You can access the different disk drives (Hard Drive, Floppy Drive, and Network Drives) inside *My Computer*.
Recycle Bin		When you delete a file or folder, Windows sends it to the Recycle Bin. You can restore the file or folder that are located in the Recycle Bin or you can permanently delete them by right clicking on the Recycle Bin and select *Empty Recycle Bin*.
My Documents		The *My Documents* folder is nothing more than a regular folder that resides on your Windows Desktop. However, it offers an easy-to-reach location where you can store and retrieve important data, and the icon is always available in Explorer Windows and on the desktop.
Internet Explorer		The *Internet Explorer* icon launches the Internet Explorer browser. The Internet Explorer browser is what you will use to access the Internet and the World Wide Web.

8.2 Features of Windows

- **Graphical User Interface:**

  Windows provide user-friendly interface to work. Its improved graphical user interface makes learning and using Windows more natural and easier for all types of users. It is more stable, customizable and efficient.

- **Start Button**

  Introduction of START button in Windows made life much simpler while there is a need to access multiple programs. It is the gateway of accessing most of the functionality available in the computer loaded with Windows. Just Click on the Start button anytime to start any programs, open or find documents, change windows settings, get Help, manage Files, maintain system, and much more.

- **Taskbar**

  As the name suggests, the Task bar provides information and access to applications that are being currently run by Windows. Using this, one can keep track of programs that have been activated and can switch between them.

- **Windows Explorer**

Windows Explorer acts as a directory browser and File Manager for Windows, and incorporates lots of additional features. It is an efficient, faster and user friendly way for locating and managing files on your computer.
Using Explorer one can easily browse through all the drives and network resources available and manage them.

- **Mouse**
Although you can use the keyboard for most actions, many of these actions are easier to perform with a mouse. The mouse controls a pointer on the screen. You move the pointer by sliding the mouse over a flat surface in the direction you want the pointer to move. If you run out of room to move the mouse, lift it up and then put it down in a more comfortable location. Mentioned below are the five basic mouse actions.
  - **Point to an item:** Move the mouse to place the pointer on the item.
  - **Click an item:** Point to the item on your screen, and then quickly press and release the left mouse button.
  - **Right-click an item:** Point to the item on your screen, and then quickly press and release the right mouse button. Clicking the right mouse button displays a shortcut menu from which you can choose from a list of commands that apply to that item.
  - **Double-click an item:** Point to the item, and then quickly press and release the left mouse button twice.
  - **Drag an item:** Point to an item, and then hold down the left mouse button as you move the pointer.

- **Shortcuts**
As the name suggests, SHORTCUTS establish a link with original programs, these are the shortest way of accessing files and other resources in Windows. Instead of traversing the full path of the program to access it, one can create "shortcuts".

- **Multitasking**
Multitasking allows the user to activate and accomplish more than one task at a time. For example, work on a document file in WORD programs, while copying file from other computer available on the network. With Windows computing environment, the user can do more than one task at a time.

- **Easy Internet Access**
One of the most useful and entirely new features in Windows is easy access to Internet. It provides built-in Internet functionality to setup link and access Internet with fewer amounts of Hardware and Software requirement. It also provides connectivity software from Microsoft Network (MSN) which allows you to stay in touch with family and friends throughout the world.
It also improves the efficiency of working on Internet with applications that support the latest Internet technologies, such as Internet Explorer, Java, and streaming audio and video support.

- **Great Gaming Platform**
Windows support rich graphics, high quality audio and video. It has all the functionality incorporated in it to support these technologies. All this is possible because of Windows compatibility with latest and hottest technologies like Plug and Play, AutoPlay, and built-in support for MIDI and digital audio and video.

- **Hardware Compatibility**

  Windows provides greater Hardware compatibility as compare to any other operating environment. It has flexibility of supporting hardware from different vendors. Its Plug and Play functionality allows you to insert the hardware card into the computer and when the computer is turned on Windows automatically recognizes and sets up the hardware.

- **Search Utility**

  Search Utility of Windows allows you to do searches by partial name, last modified date, or full text. In addition, you can save, rename, or view files from within the result pane, just like you can from Windows Explorer.

- **Help**

  Windows provides online help to accomplish a task. If the user is not sure how to perform a task, Windows Help will provide structured process for guiding on how to accomplish the task. Simply right-click on any object in the user interface (icons) and you'll get relevant descriptions about that object. Help can also be accessed from the start menu button.

8.3 Windows Desktop

When you start your computer, the desktop is the first thing you see after logging into the computer. For login you have to give *username* and *password*. A common desktop image shown in Fig.8.1, which shows the screen with icons for various programs. The desktop is the area where you work.

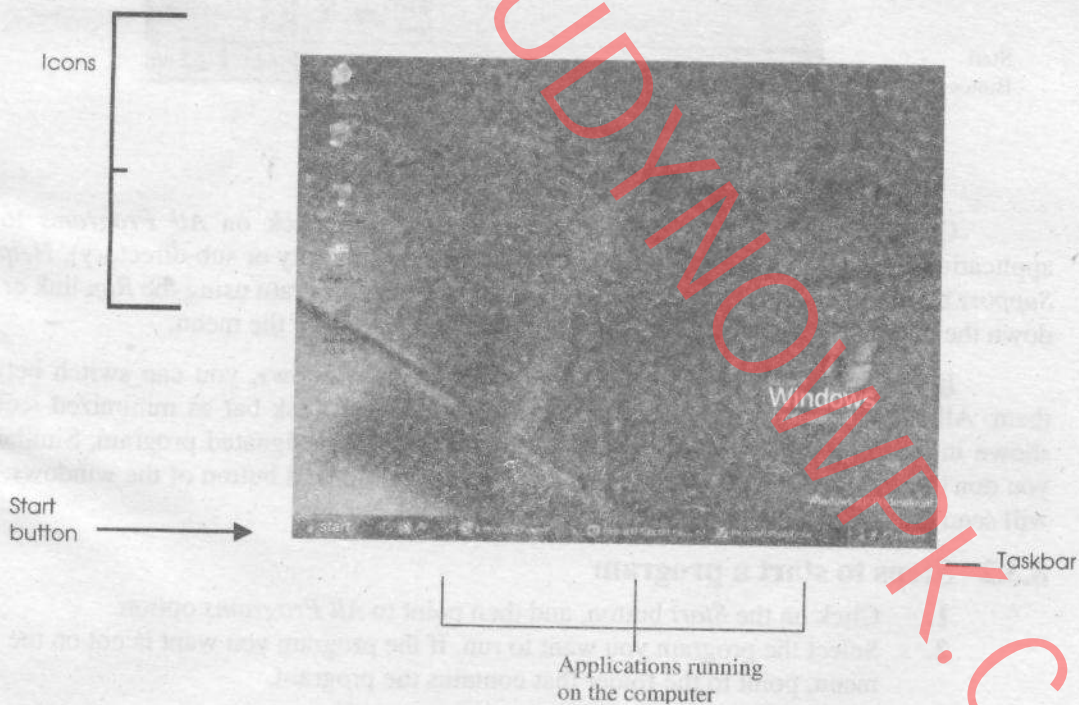



Figure 8.1: Windows Desktop

8.3.1 Start Button and Taskbar

Windows features the Start button and Taskbar at the bottom of the screen. This bar contains the Start button. Using this button, a program can be quickly started or file can be searched for. Quick help on how to use windows can also be obtained from this task bar.

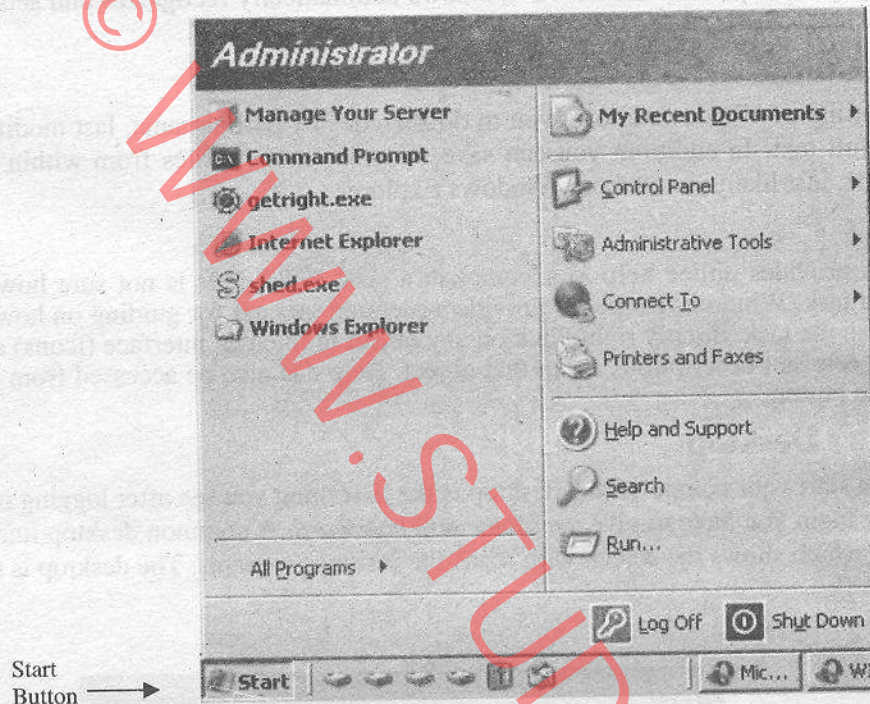



Figure 8.2: Start menu

Click on the Start to activate the Pop-up menu, click on *All Programs* to run applications, use *Search* button to find a file or folders (directory or sub-directory), *Help and Support* button to change the settings for Windows, Run a program using the *Run* link or shut down the computer using the *Shut Down* button at the bottom of the menu.

If you have opened more than one program or windows, you can switch between them. All the opened windows or program appears on the Task bar as minimized icons as shown in figure 8.1. Click on the desired icon to open the designated program. Similarly if you don't need a window too frequently, click on the minimized button of the windows. This will send the activated window as small icon to the task bar.

8.3.2 Steps to start a program

1. Click on the *Start* button, and then point to *All Programs* option.
2. Select the program you want to run. If the program you want is not on the menu, point to the folder that contains the program.
3. Click on the program icon or menu.

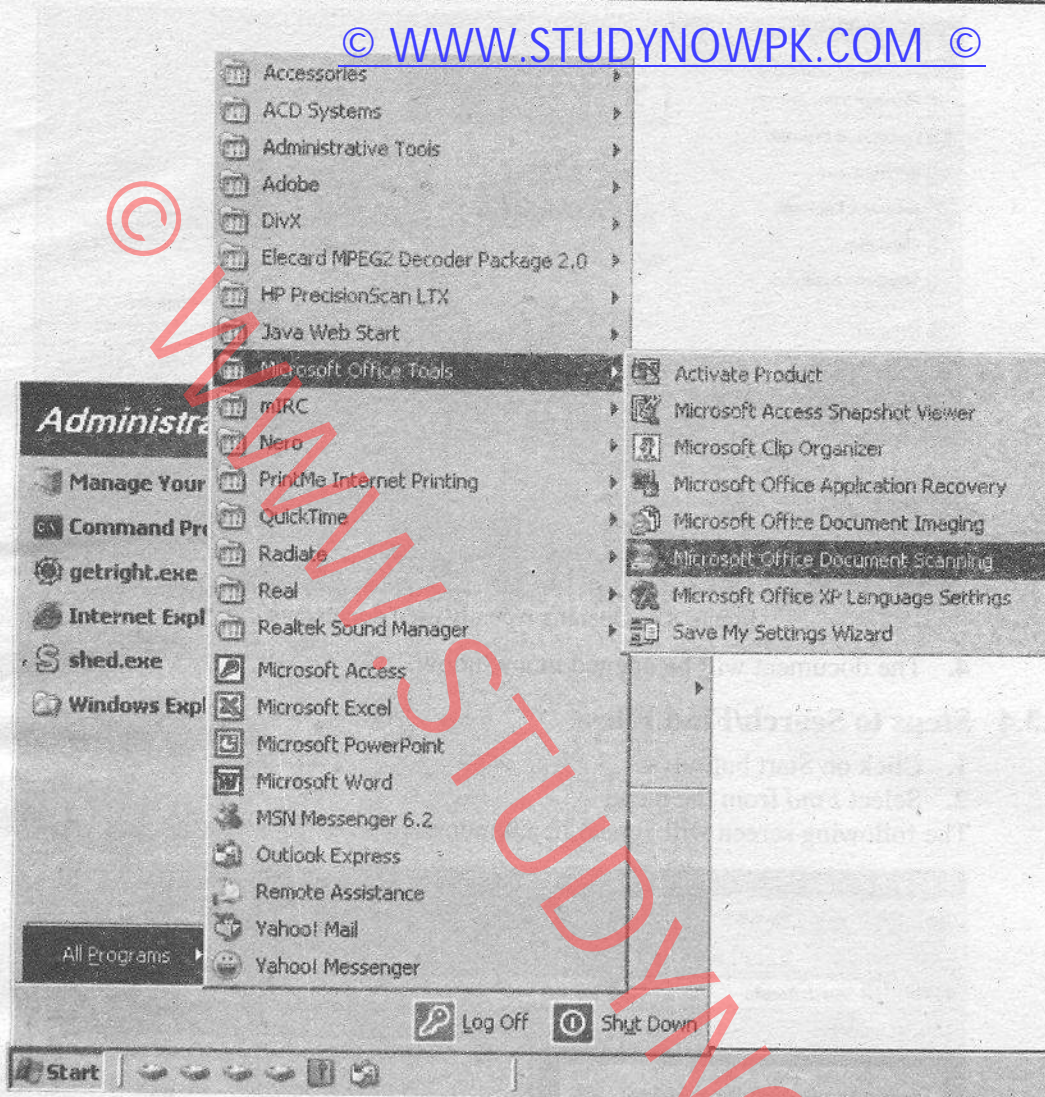



Figure 8.3: Starting a program

Once the selected program starts, a small icon appears on the taskbar. If you have opened more than one programs, click on the required program icon on the taskbar to make it currently activated program.

If the desired program is not available on the program menu or one of its submenus you can use the Find dialog box to locate the program file.

8.3.3 Steps to Open a Recently Used Document

1. Click on the Start button
2. Select/point to *Documents* option.
3. Select the document you want to open, by clicking on it

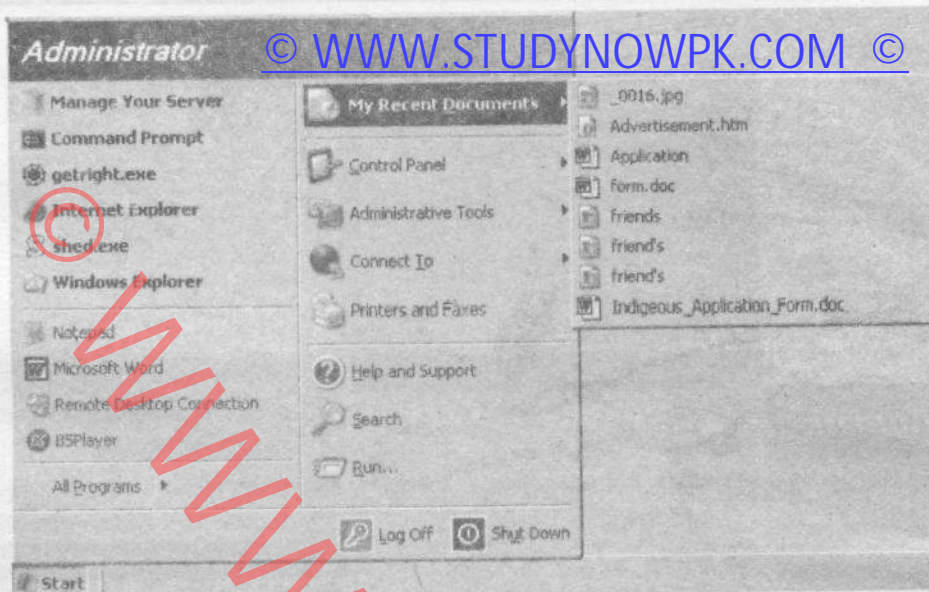



Figure 8.4: Opening recently used documents

4. The document will be opened in a window.

8.3.4 Steps to Search/Find Files

1. Click on Start button
2. Select *Find* from the menu

The following screen will appear in a window.

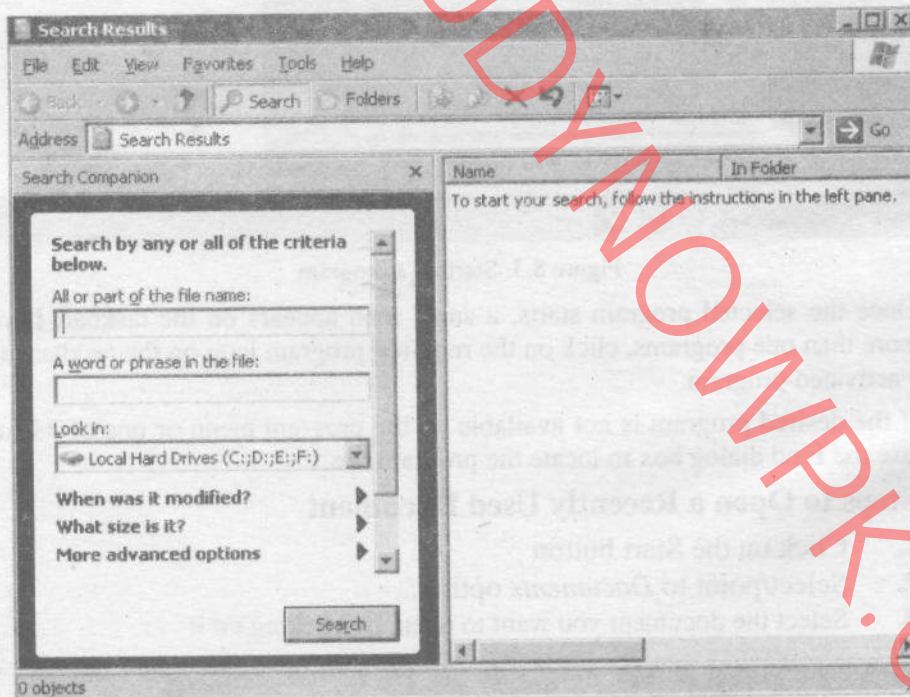



Figure 8.5: Search Window

The windows can search the file using following options.

1. Files or Folders
2. On the Internet
3. People
4. Using Microsoft outlook

The appropriate words can be entered into the textboxes and windows will start the search when you click the search button at the bottom. The Drives on your computer where the file can be found is also customizable.

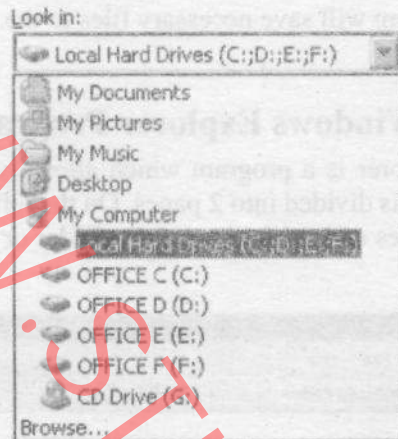



Figure 8.6: *Look in* Option in Search Window

You can ask the windows to search in a particular drive e.g. *C:* or *D:* or in all the drives by using *Local Hard Drives* option. This can be selected from the *Look In* menu. The others links available on the search pane can be used for advanced searches.

8.3.5 Steps for Shutting Down and Restarting Windows

1. Click on the Start button at the bottom left corner of screen
2. Click on Shutdown Button (on the right corner of the start menu)
3. The following box will appear on the screen

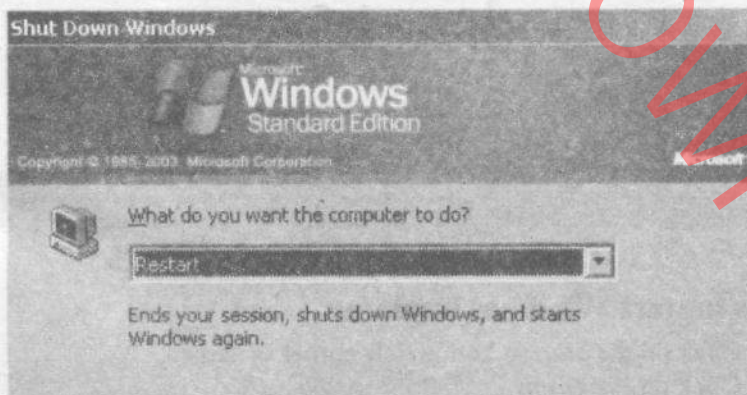



Figure 8.7: Shutting down / restarting windows

4. Select an option from the pop up menu.

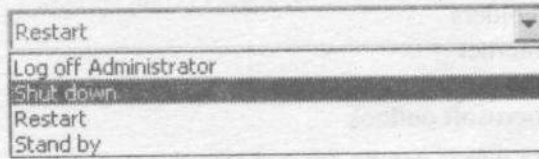



Figure 8.8: Menu for Shutdown / Restart

5. Click on OK.
6. The system will save necessary files to the disk and then take the requested actions.

8.4 Using the Windows Explorer Program

Windows Explorer is a program which allows you to view all of your folders, and files in those folders. It is divided into 2 panes. On the left side are the drives and folders. The right pane shows the files and subfolders of the folder or drive you have chosen from the left pane.

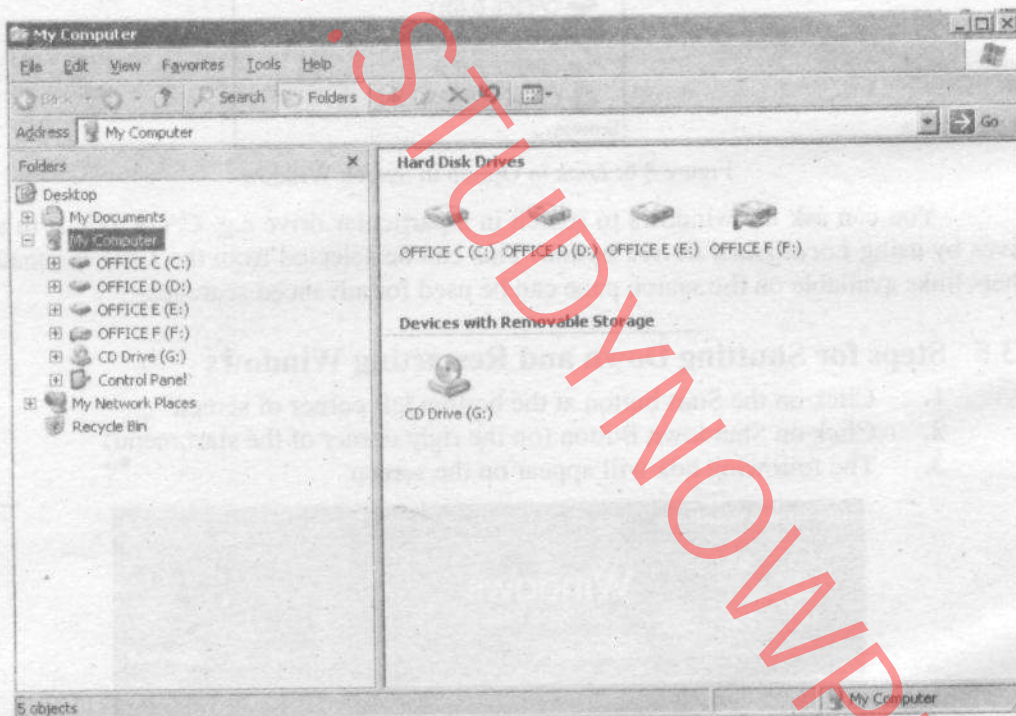



Figure 8.9: Windows Explorer Window

8.4.1 Steps to start Windows Explorer

1. Click on the *Start* at bottom left corner of screen
2. Click on *Programs*
3. Select *Windows Explorer*

8.4.2 Creating a Folder

You can create a folder to put your files in. The folder can be made in a drive inside any other folder or as a separate folder in any hard drive. To create a folder, follow these steps.

1. Select the Drive or Folder you want to create your Folder in
2. Right click in the right pane in the white area
3. Select New
4. Select Folder
5. Type in name for that Folder and press enter

Your new folder is ready for use.

8.4.3 Deleting a file or folder

1. Select the File/Folder you want to delete
2. Hit the Delete Key on the keyboard
3. Confirm Deletion

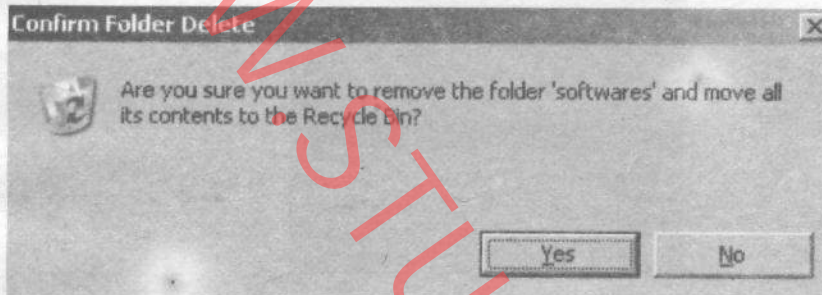



Figure 8.10: Confirming deletion of files

4. The folder and the files in the folder will be moved to the recycle bin.

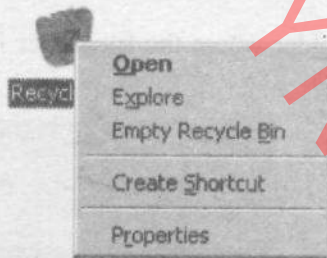



Figure 8.11: Emptying the Recycle Bin

5. They can be permanently removed from the recycle bin by right clicking on the recycle bin and selecting the option *empty recycle bin.*

8.4.4 Copying a File/Folder from a Floppy Disk to Other Drive

1. Select 3.5 Floppy (A:) in the left pane .
2. Select the File/Folder you want to copy from in the floppy drive .

3. Right Click on the File/Folder and click Copy

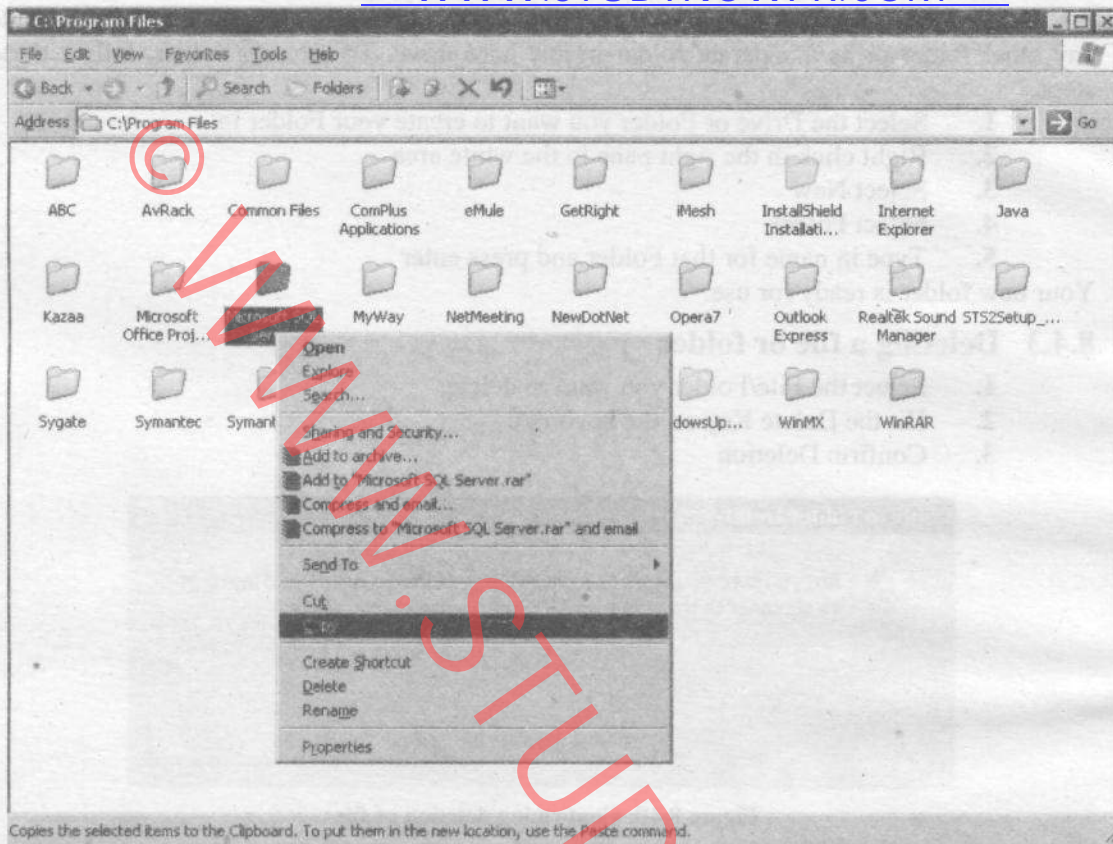



Figure 8.12: Copy file option in right click menu

4. Go to the destination folder where the file/folder is to be copied. Right click in the window in the white area

Figure 8.13: Paste option in right click menu

5. Use the paste option in the right click menu. The windows will show file copying progress box and it will automatically close when the copying is complete.

8.4.5 Renaming a File or Folder

- Select the File or Folder you want to rename
- Right Click on it

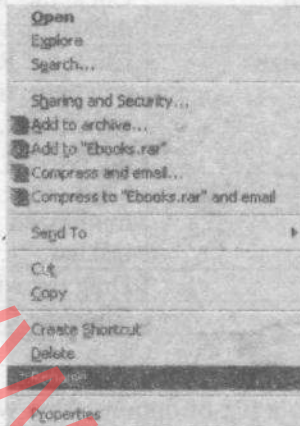



Figure 8.14: Rename option in right click menu

- Select Rename
- Type in the new name and press *Enter* button

8.5 Using Windows Controls

All programs designed for use on computers that have Windows installed have common controls that you use to scroll, size, move, and close a window. The figure below shows a typical *My Computer* and some of the common controls available in it.

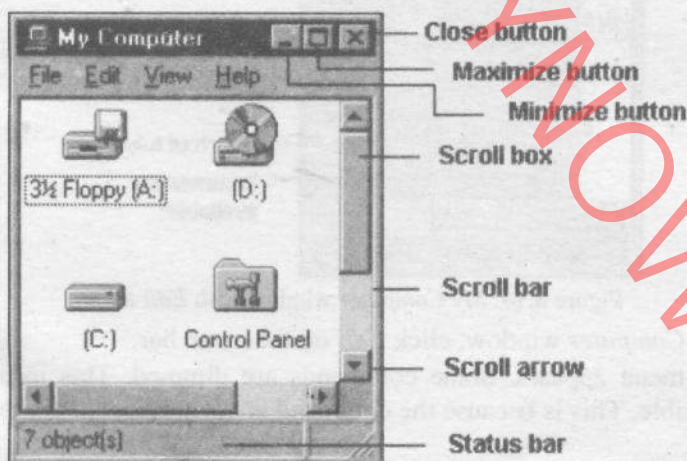



Figure 8.15: My Computer Window with controls

The table below lists a required task to be performed by a user and then method for performing the required tasks.

Required Task	How to perform the task
Move, or scroll, vertically or horizontally through the contents of a window that extends beyond the screen	Click a scroll bar or scroll arrow, or drag the scroll box. Figure above identifies these controls.
Enlarge a window to fill the screen	Click the Maximize button, or double-click the window title bar.
Restore a window to its previous size	Click the Restore button, or double-click the window title bar. When a window is maximized, the Maximize button changes to the Restore button.
Reduce a window to a button on the taskbar	Click the Minimize button. To display a minimized window, click its button on the taskbar.
Move a window	Drag the window title bar.
Close a window	Click the Close button

8.6 Using Windows Menu's

A program menu provides a list of options from which you can choose. On program menus, these options are called commands. To select a menu or a menu command, you click the item you want.

8.6.1 How to Open and make selections from a menu

1. On the Desktop, double-click the *My Computer* icon.

   The *My Computer* window opens.

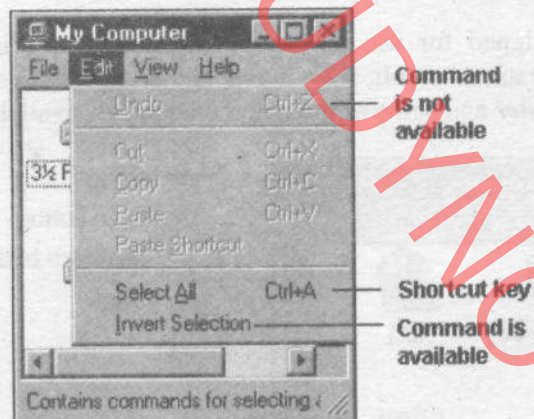



Figure 8.16: *My Computer* window with *Edit Menu*

2. In the *My Computer* window, click *Edit* on the menu bar.
   The *Edit* menu appears. Some commands are dimmed. This means the command isn't available. This is because the command is not applicable to the current selection or option.
3. Click the *Edit* menu name to close the menu.
   The menu closes.
4. Click *View* on the menu bar to open the *View* menu.
5. On the View menu, click List.
   The items in the My Computer window now appear in a list, rather than as icons.

6. On the View menu, point to Arrange Icons By.
A cascading menu appears listing additional menu choices. When a right-pointing arrow appears after a command name, it indicates that additional commands are available.
7. Click anywhere outside the menu to close it.
8. Click the Close button in the upper-right corner of the My Computer window to close the window.

8.7 Changing Desktop Background

To change the background of the desktop
1. Right Click on the Desktop.
2. Click on *Properties*

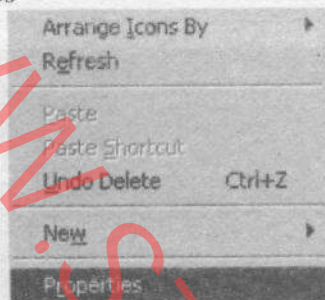



Figure 8:17 Selecting Desktop properties

3. In the Box that appears, click on *Desktop* Tab and choose a background.

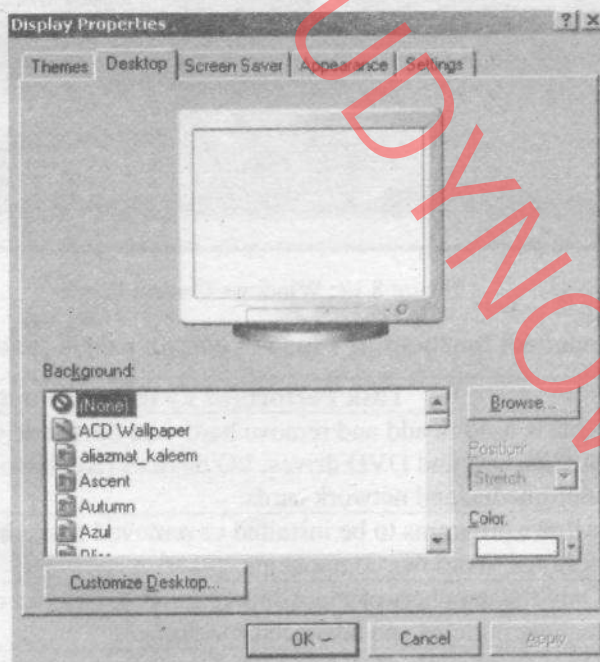



Figure 8.18: Changing Windows Background

4. Click on *Apply* and then click *Ok*.
Similarly all other option such as Screen Saver and Settings can be used from their respective tabs.

8.8 Using Control Panel

The Control Panel allows you to customize features such as "Appearances and Themes" or "Printers and Other Hardware". Additional features guide you further. For example, if you are downloading pictures from a digital camera for a class project, selecting "Printers and Other Hardware" provides you with the "Scanners and Cameras" option. By choosing the category you want, you can complete your task easily.

If you prefer the classic Windows control panel, that option is just a mouse-click away.

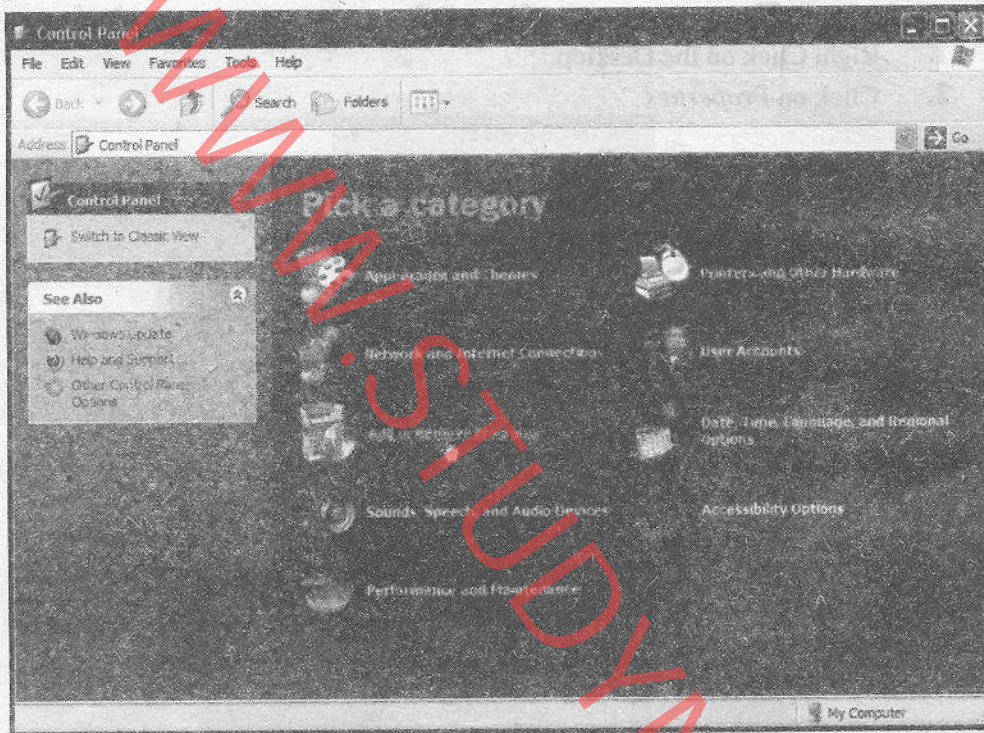



Figure 8.19: Windows Control Panels

Some of the important functions of windows control panels are described below.

Option	Task Performed by using the option
Add/Remove Hardware	This is used to add and remove hardware device drivers for devices such as CDROM and DVD drives, I/O devices (Keyboard, mouse), modems, multimedia, and network cards.
Add/Remove Programs	Allows programs to be installed or removed from the system including optional Windows components.
Administrative Tools	Only the members of the Administrators group can use these tools for making policies and advanced functions.
Display	Used for changing Background, Screen Saver, Appearance and Settings.
Internet Options	These are options are for Internet Explorer. Used for configuring advanced options of internet explorer.
Keyboard	For configuring keyboard speed and other attributes
Mouse	For setting mouse pointer and mouse speed

Printers	Allows Installation and Removal of printers from the computer. It can also be used to search printers from the network.
Sounds and Multimedia	This can be used to setup sound schemes and sounds to play for specific events.
Scheduled Tasks	This is also known as "Task Scheduler", it is used to schedule programs to run at specific times. An "Add Scheduled Task" icon is present in this folder which can be used to add new task.
Game Controllers	Allows configuration of joysticks and game pads for computer games.
Fonts	This allows viewing of current fonts and installation of new fonts for writing text. It is a shortcut to the font's folder.
Scanners and Cameras	For installing and configuring Digital cameras

8.9 Computer Virus and Antivirus

8.9.1 Computer Virus

A **computer virus** may be a program or a set of programs that can cause extensive damage to your computer system.

This code is attached to some part of the normal computer operating system or computer program. Instructions in this code tell the computer to perform some task. This task is often a destructive one, such as deleting important information or crashing the hard disk. However, there are other viruses that may only slow down your computer or do no serious damage. Some viruses have been known to do nothing more than put a large happy face on your computer screen.

8.9.2 How Does a Computer Get a Virus?

Just as a biological virus is passed from person to person, a computer virus is passed from computer to computer. A virus can be attached to any file that you copy to your computer. If you download files from the Internet or copy programs or files on floppy disks, you are very susceptible to viruses. Whenever you download files or put a floppy disk into your computer, you are susceptible to viruses.

Many viruses are spread through e-mail. Generally, you cannot get a virus from simply reading e-mail. Certain types of today's viruses, like Klez, are different. They are very dangerous because you do not even have to open an e-mail attachment to release the virus. Just opening the e-mail itself can release the dangerous bug.

Under normal circumstances, a virus becomes active when you execute a program that contains the virus. For instance, if you download a program from the Internet and it is infected with a virus, the virus will attack your computer when you activate the program.

8.9.3 Antivirus

Antivirus is software that is used to detect and remove a virus on a computer system.

You can stop viruses before they enter your computer. You can do this by using a good virus protection program i.e. anti virus software. it will check all files for viruses. Once installed, an antivirus program can be set to work in the background. It will check all files before they enter your computer and will alert you if a virus is detected before it contaminates your system. If a virus is detected, your antivirus program will quarantine or eliminate it so that it cannot harm your computer.

Be very careful to use a virus protection program that matches your operating system. If you use a Mac, use a virus protection program made especially for Macintosh computers. If you use Windows 98 or XP, select a program that was written specifically for your operating system. Don't try to use a virus protection program that was written for Windows 95 on a Windows XP machine; this will cause unstable behavior and may cause your computer to crash or malfunctions.

New viruses are written everyday. Companies like McAfee, Symantec, and Panda Software are constantly expecting new viruses and updating their antivirus software tools.

Exercise

1. Briefly explain the following:
   a. Drives b. Folders c. Directory d. File extension e. Icon
2. Name and explain three different features of Windows.
3. What is start button and taskbar?
4. Write the steps needed to start a program?
5. What are the steps for shutting down and restarting Windows?
6. What is Windows explorer? How can We start Windows explorer?
7. What is the use of recycle bin?
8. What is "Control Panel"? Name three different options of control panels and tasks performed them.
9. Write a note on computer virus.
10. What are the benefits of using anti-virus programs?
11. **Fill in the blanks :**
    (i) Windows dominates the personal computer world, running almost on ________% of all personal computers.
    (ii) in the file name PhoneNumbers.txt, the part "PhoneNumbers" is the ________of the file and ".txt" is the ______of the file
    (iii) When you delete an object, Windows sends it to the ____________
    (iv) ____________ acts as a directory browser and File Manager for Windows
    (v) Windows places the deleted files and folders in ____________
    (vi) ____________ is just a link to the original file/folder or an application.
    (vii) ________can seriously damage the computer system.
    (viii) Windows is a(n) ____________
    (ix) A(n) ________ is a software that is used to remove the viruses from the system.
    (x) To add or remove hardware, you must select an option from ____________
12. True or False
    (i) A Network is a group of Computers connected to each other to share available resources such as storage and printers.
    (ii) Windows do not provides online help.
    (iii) Multitasking allows the user to activate and accomplish more than one task at a time.

(iv) Folders are used to organize the data stored on your drives.
(v) Icons help you to remember file names in windows.
(vi) The Internet Explorer browser is what you will use to access the Internet and the World Wide Web.
(vii) Windows is just a GUI, not an operating system.
(viii) shift + ctrl + delete command sequence is recognized only by Windows.
(ix) Contents of Recycle bin can be restored to.
(x) NORTRON antivirus software can detect all viruses in the computer.

13. Choose the correct answer

i. ____________ the gateway of accessing most of the functionality available in the computer loaded with Windows
   a. Taskbar
   b. Folders
   c. Start Button
   d. Mouse
   e. Application

ii. Control Panel allows you to customize which of the following feature:
   b. Appearances
   b. Themes
   c. Printers
   d. Other Hardware
   e. All of the above

iii. Computer virus is simply a
   a. a disease
   b. set of computer instructions or computer code
   c. type of bacteria
   d. hardware component
   e. none of the above

iv. Klez is a
   b. Game name
   c. Hardware component
   d. Virus name
   e. Person name
   f. None of the above

v. Good Times was a
   a. virus
   b. anti virus
   c. fake news about the existence of virus
   d. none of the above

Answers

Q.11
(i) 90 (ii) Name, Extension (iii) Recycle bin (iv) Windows Explorer
(v) Recycle Bin (vi) Shortcut (vii) Virus (viii) Operating system
(ix) Antivirus (x) Control Panel

Q.12
(i) T (ii) F (iii) T (iv) T (v) F (vi) T (vii) F (viii) T (ix) T (x) F

Q.13
(i) c (ii) e (iii) b (iv) c (v) c

Index

GLOSSARY

Term	Definition
Address Bus	An Address Bus is a set of wires similar to the data bus but it only connects central processing unit (CPU) and memory.
Alphabet data	Data consisting of only language (English) characters is called *alphabet data*.
Alphanumeric Data	Data consisting of alphabets of a language, numbers and special characters is called *Alphanumeric Data*
ALU	Arithmetic and Logic Unit
Antivirus	Antivirus is software used to detect and remove a virus on a computer system.
Application Software	Computer programs designed to accomplish some specific user's tasks are called *Application Software*.
ASCII	American Standard Code for Information Interchange
Assembler	A program for converting an assembly language program into machine language is called *Assembler*.
BASIC	Beginners All Purpose Symbolic Instruction Code
BCD	Binary Coded Decimal
Bit	Binary Digit: the smallest unit of information.
Bit Rate	The number of bits transferred per second is called *bit rate*.
Boolean Constants	The values 0, 1 are the *Boolean constants*.
Boolean expression	An expression consisting of Boolean constants, variables and operations is called *Boolean Expression*.
Boolean variables	The variable that can take only one of the two values (i.e. 0 or 1) is called *Boolean Variable*
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer Aided Manufacturing
CD	Compact Disk
CDR	Compact Disk Recordable
CD-ROM	Compact Disk-Read Only Memory
CDRW	Compact Disk Re-Writer
Code	An arrangement of symbols to stand for a word or an action is called *code*.
COBOL	Common Business Oriented Language
Compiler	A program for converting a high level language program into machine language is called *Compiler*
Computer	A *computer* is an electronic device that processes data according to the given instructions and converts it into information.
Control Bus	The *control bus* carries control information from the control unit to the other parts of the computer.
CPU	Central Processing Unit
CRT	Cathode Ray Tube
DRAM	Dynamic Random Access Memory
Data	Raw facts and figures to which no meaning is attached and it is ready for processing are called *data*.
Data Bus	A *data bus* is an electrical path that connects the central processing unit (CPU), memory, input / output devices and secondary storage devices.
Data rate	The rate at which data is delivered to the CPU by a device is called *data rate*.
Decimal Number System	A method of representing information using 10 symbols(0,1,2,3,4,5,6,7,8 and 9) is called *decimal number system*.
Digital	A discontinuous or discrete entity is called *digital*.
Digital Data	Data represented by discrete values or conditions is called *digital data*.
Disk Formatting	Preparing a storage disk for storing data by dividing it into tracks/sectors is called *disk formatting*.
DRAM	Dynamic Random Access Memory – Contents of this type of memory needs to be refreshed periodically.
Duality principle	The Principle of Duality states that any result deduced from the axioms of Boolean algebra remains valid if the following steps are performed

	▪ All 0's in the result are changed to 1 and vice versa.
	▪ The "." in the original result is changed to + and vice versa
Dumb Terminal	A terminal that has no processing capabilities is called *Dumb terminal.*
EBCDIC	Extended Binary Coded Decimal Interchange Code
EDC	Electronic Digital Computer
EDP	Electronic Data Processing
EEPROM	Electrically Erasable Programmable Read Only Memory
Electronic Mail(e-mail)	A method of sending messages electronically based on mailbox addresses is called *electronic mail.*
ENIAC	Electrical Numerical Integrator And Calculator
EPROM	Erasable Programmable Read Only Memory
FDD	Floppy Disk Drive
FORTRAN	Formula Translation
GB	Giga Bytes (2^{30} bytes)
GUI	Graphical User Interface
Hard Copy	Printed version of a document on a paper is called hard copy.
Hardware	Physical parts of the computer that can be seen and touched are called *hardware.*
Hertz (Hz)	Unit of measurement of frequency
Hexadecimal Number System.	A method of representing information using 16 symbols(0,1,2,3,4,5,6,7,8, 9 and A, B, C, D, E and F)
IBM	International Business Machines
IE	Internet Explorer
Impact Printer	An *impact printer* produces images by striking an inked ribbon with a hammer or a set of pins, pressing ink from the ribbon onto a piece of paper.
Information	Processed data to which a clear meaning is attached is called *information.*
Intelligent Terminal	A terminal that have its own processing power is called *intelligent terminal.*
Interpreter	A program for translating a high level language program into machine language, one line at a time is called *interpreter.*
I/O	Input/ Output
ISO	International Standards Organization
JVM	Java Virtual Machine
KB	Kilobytes (2^{10} bytes)
LCD	Liquid Crystal Display
LISP	List Processing Language
Logic Gate	An electronic circuit that take one or more inputs and produces one output is called *Logic Gate.*
LPM	Line Per Minute
LPT	Line Printer
LSB	Least Significant Bit
LSI	Large Scale Integrated Circuit
Maxterms	Sum of literals is called a *maxterm.*
MB	Mega Bytes (2^{20} bytes)
Microprocessor	*Microprocessor* is a complete processing circuitry on a chip. It is also called brain of the computer
Minterms	Product of literals is called a *minterm.*
MSB	Most Significant Bit
MSN	Microsoft Network
Multitasking	*Multitasking* allows the user to activate and accomplish more than one task at a time.
Numeric Data	Data consisting of numbers only is called *numeric data.*
Octal Number System	A method of representing information using 8 symbols(0,1,2,3,4,5,6,and 7) is called *octal number system*
OCR	Optical Character Recognition
OOP	Object Oriented Programming

Operating system	An *operating system* is a set of programs running in the background on a computer system and providing an environment in which other programs can be executed and the computer system can be used efficiently.
OS	Operating System
PC	Personal Computer
PDA	Personal Data Assistants
Parallel port	A *parallel port* allows transmission of more than one bits (e.g. 8 or 16 bits) of information at a time.
Port	A *port* is a socket that enables an external device such as a printer to be attached to the computer
Precedence	The order in which operators are evaluated is called precedence.
Processing	*Processing* means to manipulate, calculate, distribute or arrange data/information
Program	A *program* is a set of instructions given to the computer to solve a particular problem
PROM	Programmable Read Only Memory
RAM	Random Access Memory
Register	High speed memory locations built inside the CPU to temporarily hold the data for processing is called *register*.
Resolution	The *resolution* refers to the number of pixels on the screen, expressed as a matrix.
ROM	Read Only Memory
Serial Access	Accessing data in linear order is called *serial access*.
Serial Port	A serial port allows transmission of one bit of information at a time
Seek time	The time used to move the head to the appropriate track after reading the address is called *seek time*.
Signed Number	A representation of binary numbers including sign(plus or minus) is called *signed number*.
Simulation	*Simulation* is the use of computer to represent the dynamic responses of on system by the change in behaviors of another system modeled after it.
Softcopy	Electronic version of a document is called softcopy.
Software	A set of instructions given to a computer for solving a problem is called *software*.
SRAM	Static Random Access Memory – Contents of this type of memory do not need to be refreshed.
SVGA	Super Video Graphic Array, has resolution 1024x768
System Bus	A *bus* or *system bus* is composed of a set of communication lines or wires, and connects different devices to communicate each other.
System Software	Computer Programs that are responsible for controlling and managing the task overall working of the computer is called *system software*.
Terminal	A monitor and keyboard connected to a mainframe computer
Truth table	A table showing output for all possible inputs is called *truth table*.
USB	Universal Serial Bus
UNIVAC	Universal Automatic Computer
VB	Visual Basic
VGA	Video Graphic Array has resolution 640x480 pixels.
Virus	A *virus* is a program or a set of programs that can cause extensive damage to your computer system.
VLSI	Very Large Scale Integration
Von-Neumann Theory	Data and instructions(program) resides in the same read/write memory. Thus the machine can itself alter either its program or internal data.
Web Page	A hyperlinked document available on WWW is called *Web Page*.
Web Site	A collection of interlinked web pages is called *web site*.
WWW	World Wide Web